ROBUST NON LINEAR FILTERING AND ADAPTIVE MANIPULATOR CONTROL

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

by
RAVINDRABABU KONERD

to the
DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
MARCH, 1990

CERTIFICATE

Certified that the work embodied in this thesis entitled " **Robust Nonlinear Filtering and Adaptive Manipulator Control** " has been carried out by Mr Ravindra Babu. K. under my supervison and the same has not been submitted elsewhere for a degree

Dr Arindam Ghosh
Department of Electrical Engineering
Indian Institute of Technology
KANPUR

ACKNOWLEDGEMENTS

I am very grateful to my guide Dr. Arindam Ghosh for his continued guidance and encouragement throughout the course of this work. I am indebted to Dr. R.N.Biswas for allowing me to use his PC. My heartful thanks to all my friends who made my stay here, a memorable one.

Ravindra Babu. K.

CONTENTS

CHAPTER 1

Introduction

A robot manipulator is a computer controlled mechanical arm which ca perform various tasks. These are gaining wide acceptance particularly i industrial automation like automated manufacturing, etc With rapid expansion in th field of robotic manufacturing systems, the problem of control of robot ar motion has become the subject of considerable research interest in the recer years.

The robot control problem is to determine the joint torques that produce desired motion of the joints. Most of the control schemes presently available ar based on independent joint control scheme, in which each joint is independentl controlled by simple PID type of control law with predefined constant gains. These control schemes may be adequate for simple pick and place tasks. However, many o the modern day robots are required to operate under varying circumstances, such as loading and with high speeds. Hence, the existing control schemes are severely inadequate.

1.1 Review of Literature :

During the last decade, with the development of Adaptive Control theory, many authors made significant contributions to the field of control of robots by formulating adaptive laws and demonstrating their usefulness through simulations and experiments.

One of the early contributions to the field of adaptive control of robot manipulators came from Horwitz and Tomizuka [1]. Their scheme assumed that plant inertia is time invariant and assumed gravity, friction torques to be known With

these assumptions, they formulated each joint as a second order Single inp Single Output (SISO) system. Then for each degree of freedom they used I controller with an additional feedforward term to compensate for the friction a gravity forces But with this formulation this system stability can not be assur under parameter variations.

Another approach to the manipulator control problem can be classified a global linearization of the robot dynamics. Arimito and Takegaki [2] obtain adaptive control algorithms based on local parameter optimization for a robo described by a linear time varying model derived from the linearization around th desired trajectory. This method assures stability of the error system. Howeve they assumed low speed of motion and neglected centrifugal and Coriolis force The simulation results with a nonlinear robot model produced tracking errors wit a maximum of 1 cm when the trajectory is a straight line with a maximum speed o 0.1 m/s.

Many of the later works in this area applied adaptive control theor developed for linear time invariant systems by making approximations about th nonlinear model. The highly coupled nonlinear equations of the robot model ar linearized about the planned trajectory to obtain a perturbed system. Th adaptive control law is based on the linearized perturbed equations about th reference trajectory which takes into account all the interacting forces betwee the joints. Lee and Chung [3] used recursive least squares to identify th perturbed system and used one step ahead optimal control law to stabilize th perturbed system. But when the perturbations are large such as load changes thi control scheme may give large errors.

M.H.Liu [4] developed a control scheme using generalized least square self-tuning controller with pole assignment. For each joint an independent control scheme is proposed using linearized dynamic equations. Various interaction

between the joints are neglected which result in poor performance at high speeds and also the loading effects are not considered. The simulation results show that the trajectory deviation is large in the begining of the trajectory due to improper selection of the initial values. However, initial values play a very significant role in the parameter convergence of such schemes. This has been dealt with some detail by Waknis [5] and Ghosh et al [6]. They showed that linearized models in general do not guarantee a minimum phase behaviour, and choosing of control parameters by trial and error may lead to severe divergence .

Many control algorithms have also been developed which assume the full nonlinear manipulator model in the compensation These results belong to one of the two general categories : " *adaptive high-gain cancellation of nonlinearities* " and " *adaptive computed torque* ". Of the first type, notable contributions include Balestrino [7] and Oshima et al [8]. These adaptive schemes are similar in that they utilize tracking error feedback (position and rate errors) with an adaptive gain that grows until it is large enough to cancel the nonlinearities. An unanswered question in these schemes is how to select certain gains and parameters within the adaptation loop which depend upon the closed loop plant. Another major drawback with many of these schemes is that the control may chatter and excite the unmodelled dynamics ; and the robustness to these effects has not been established.

The second category of the existing adaptive controller designs with full nonlinear manipulator plant model utilize the computed torque control law with some additional terms to provide for stability Basically, the computed torque technique uses the full dynamic compensation with feedforward and feedback components. The feedforward control components compensate for the interaction forces among various joints and the feedback components compute the necessary correction torques required to compensate for any deviation from the desired trajectory. Contributions in this area include Craig, Hsu and Sastry [9], Middelton

and Goodwin [10], Slotine and Lee [11], [12] and A.J.Goldenberg etal [13]. In [11] load and friction parameters are estimated as a part of a procedure whose aim is to force the error between the desired and actual joint coordinates to zero. This paper's main contribution is a theortical result. The proof of stability of the proposed globally convergent procedure is given. The robustness of these adaptive computed torque schemes with respect to plant uncertainties such as unmodeled dynamics, parameter variations has not been analyzed.

All the above control schemes discussed assume noise free feedback measurements (of positions and velocities of joints) and no process (robot model) noise. But none of these requirements are fulfilled for a typical robot manipulator due to the relatively low resolution of the sensors employed. Blauer and Belanger [14] attempted to solve this problem by using the Extended Kalman Filter theory. They expressed the noisy on-line state and force measurements as functions of state and surface parameters. The Extended Kalman Filter is then used to give the optimal estimates of the states. Their simulation results show good convergence property and relative insenstiveness to increasing noise intensities. But they assumed that the robot dynamics is precisely known . Majee [15] solved this problem by estimating the robot dynamics using the tracking error information.

Thus a large volume of literature available dealing with this subject vary on the basis of concept or implementation. Most of the methods made their control scheme robust by utilizing various assumed dynamic and kinematic models These models are typically inexact containing uncertainity in parameters. Moreover, most of the methods discussed above used adaptation as a means to simplify or otherwise to avoid performing complex computations in the control loop. As computing power becomes more readily available with the advent of fast microprocessors this motivation diminishes.

1.2 Scope of Work :

The objectives considered here for the control of the robot manipulators are

i) The control scheme should be insensitive to noisy measurements of the system states.

ii) The deviations from the desired trajectory i.e., the tracking errors should be small and convergence should be fast.

iii) The control scheme should be robust enough to reject uncertainity in dynamics and its performance should be same at all operating conditions.

Keeping in mind all the above objectives a control scheme based on Linear Quadratic Regulator theory is proposed. The basic control scheme consists of an identifier based on Recursive Least Squares procedure to identify the robot dynamics, a feedforward term computed from the identified dynamics to compensate for the various interaction forces among the joints and a feedback term computed from the identified dynamics to compensate for the model variations and parametric uncertainities. The feedback term is based on optimal control law derived from the linearized perturbation equations. This is coupled with the state estimator based on Kalman Filter theory to filter out any noise corrupting the feedback system. The importance of the identifier in the control scheme is demonstrated by comparing the performance of this scheme with the scheme without identifier in the control loop. Also, the performance of these two control schemes with two different state estimators, Extended Kalman Filter and Constant Gain Extended Kalman Filter is evaluated.

1.3 Thesis Layout :

In Chapter2 the different control blocks which, when integrated constitute the full control scheme are introduced. These consist of a controller based on Linear Quadratic Control theory, and two state estimators based on Kalman Filter theory to filter out any noise corrupting the feedback measurements. The nonlinear continuous time model of the two degree of freedom planar manipulator model and the state trajectories used for simulation and analysis purposes are also included.

Chapter 3 discusses one of the two control schemes presented in this thesis. This control scheme consists of feedforward term to compensate for the various interaction forces among the joints, which can be computed off-line, and a feedback component based on LQ theory to compensate for the model variations and parametric uncertainites The noisy feedback measurements are filtered using the estimation schemes mentioned in chapter2 The performance of the controller in conjunction with each of the estimators is studied. Finally, the simulation results referred to the simulated robot model are presented.

Chapter 4 provides the concept of Adaptive Linear Quadratic Control. Here the robot dynamic parameters such as load, masses of links etc. are estimated on line using standard recursive least squares technique and the feedforward term is calculated from these identified dynamic parameters. The performance of this adaptive controller in conjunction with each of the state estimators is studied using the simulated robot model. The simulation results and comparative performance evaluation of the two controllers are presented .

In Chapter 5, two control schemes based on Computed Torque Control theory, which use full nonlinear feedback compensation are discussed. In the first scheme [15] the parameter perturbation effects are compensated by using sliding mode control. In the second scheme, which is essentially a modification of [11], these effects are compesated by adapting the dynamic parameters as a part of the

procedure to make closed loop system asymptotically stable. The results of the simulation experiments carried on the two degree of freedom planar manipulator are also included.

In chapter 6 the performance of all the control schemes discussed in th this thesis are evaluated and the thesis is concluded with the summary of ideas which have been presented.

CHAPTER 2

Basic Control Blocks

This Chapter deals with the various basic ideas which will be integrated later to derive the complete robot control scheme. The first of these ideas deals with the mathematical formulation of the dynamic equations of the robot arm motion which are useful for computer simulation of robot motion and in design of suitable control equations. These equations are derived for a two degree of freedom planar manipulator. This discussion is followed by a trajectory generation scheme through which the cartesian trajectory of the two degree of freedom planar manipulator is converted into its equivalent joint coordinates. The state space representation of the dynamic equations is derived which facilitates the control law formulation. Next, two global state estimation schemes to filter out the noisy state measurements for state feedback are discussed. The details of the linearized perturbation equations derived and the control scheme based on Linear Quadratic Regulator theory are also included .

2.1 Dynamics of Robot Manipulator :

The closed form dynamic equations describing the robot arm motion are given in the vector-matrix notation [16] as

$$\underline{u} = D(\underline{q})\ddot{\underline{q}} + H(\underline{q}, \dot{\underline{q}}) + g(\underline{q}) \qquad (2.1)$$

where $\underline{q}$, $\dot{\underline{q}}$, $\ddot{\underline{q}}$, $\underline{u}$ are n dimensional joint position, velocity, acceleration and joint torque vectors, respectively; $D(\underline{q}) \in \Re^{n \times n}$ $\forall$ $\underline{q}$ is the manipulator moment of inertia matrix; $H(\underline{q},\dot{\underline{q}}) \in \Re^{n \times 1}$ $\forall$ $\underline{q}$, $\dot{\underline{q}}$ represents the centrifugal and Coriolis forces acting on the joints; $g(\underline{q}) \in \Re^{n \times 1}$ $\forall$ $\underline{q}$ represents the gravitational loading on the joints.

In the derivation of eq. (2.1), the actuator dynamics, the terms due to

friction, and effects due to torque saturation are neglected. Even with these simplifications, it is seen that the underlying modes of eq. (2.1) are nonlinear and strongly coupled. The dynamic model, eq. (2.1) forms the starting point for all the control efforts described in the following pages

2.2 Robot - Simulation :

A two-degree of freedom planar manipulator is chosen for the experiments performed in this work. The manipulator configuration is shown in fig. (2.1)

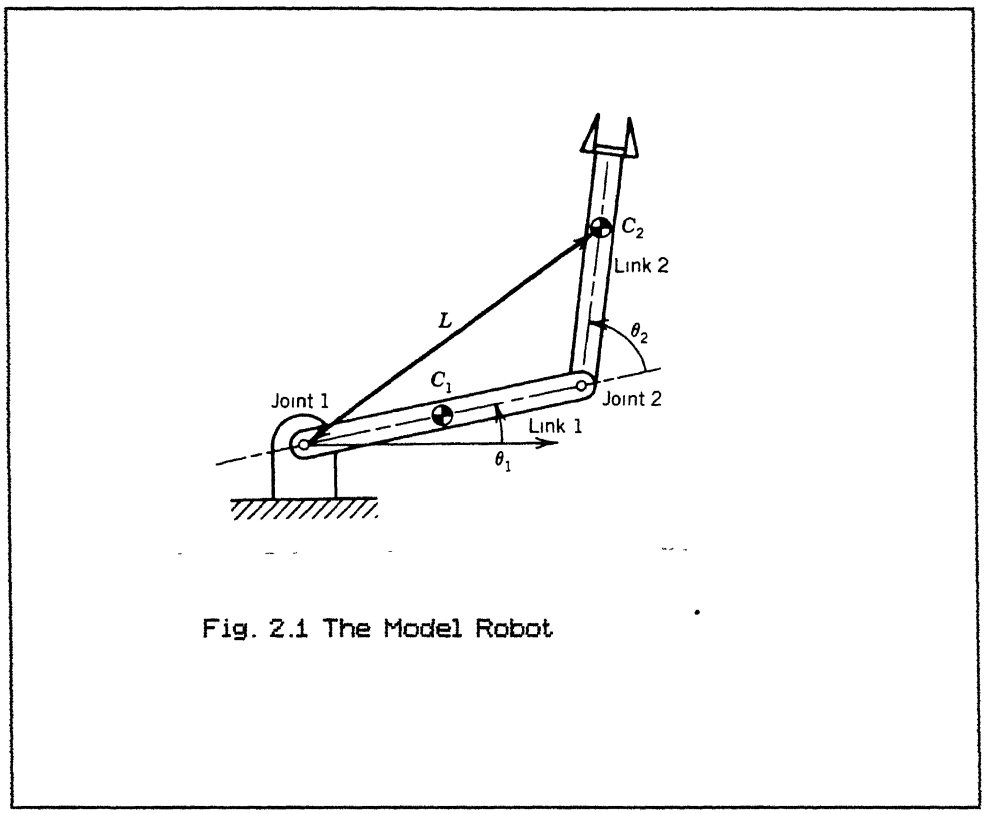


Fig. 2.1 The Model Robot

For this two- degree of freedom manipulator, the total number of joints, n is equal to 2. Under this condition, the torque and position vectors are given by

$$\underline{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad , \quad \underline{q} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \qquad (2.2)$$

and the D, H, and g matrices are given by

$$D(\underline{q}) = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}, \quad H(\underline{q}, \underline{\dot{q}}) = \begin{bmatrix} H_{11} \\ H_{21} \end{bmatrix} \text{ and } \quad g(\underline{q}) = \begin{bmatrix} g_{11}(q) \\ g_{21}(q) \end{bmatrix} \qquad (2.3)$$

These matrices are then written in terms of the robot link parameters (defined in Table (2.1)) as

$$D_{11} = m_1 lc_1^2 + m_2(l_1^2 + lc_2^2 + 2\, l_1 lc_2 \cos(q_2) + I_2 + I_2$$

$$D_{12} = m_2 l_1 lc_2 \cos(q_2) + m_2 lc_2^2 + I_2$$

$$D_{21} = D_{12}$$

$$D_{22} = m_2 lc_2^2$$

$$H_{11} = -m_2 l_1 lc_2 \sin(q_2)\, \dot{q}_2^2 - 2\, m_2 lc_2 l_1 \sin(q_2) q_1 \cdot q_2 \qquad (2.4)$$

$$H_{21} = m_2 l_1 lc_2 \sin(q_2) \dot{q}_2^2$$

$$g_{11} = m_1 lc_1 g \cos(q_1) + m_2 g (lc_2 \cos(q_1+q_2) + l_1 \cos(q_1))$$

$$g_{21} = m_2 lc_2 g \cos(q_1+q_2)$$

Table 2.1

link	Mass (Kg)	Length of the link i (m.)	Distance of c.m of link i from i-th coordinate frame(m.)	Inertia of Link i (m)
1	$m_1 = 15$	$l_2 = 1.4$	$lc_1 = 0.5$	$I_1 = 3.9765$
2	$m_2 = 9.6$	$l_2 = 0.8$	$lc_2 = 0.2$	$I_2 = 0.63386$

The robot model is then simulated in digital computer by solving the second order non-linear differential equation (2.1). The solution here lies in finding the joint position, velocity and acceleration vectors along a specified trajectory. To slove for these quantities the torque vector along this trajectory is given as an input. Many standard numerical methods like Runge-Kutta, etc. can be used to solve this problem. The method adopted in this thesis is similar to the one described in [15], [17] where a detailed implementation algorithm is also given.

2.2.1 Trajectory Generation :

The nominal trajectory for the robot, i.e., the reference trajectory through which the robot must travel is specified in terms of the desired end-effector velocity, acceleration and the end effector displacement with respect to world coordinates. The problem of trajectory generation is to convert these values in terms of joint positions, velocities and accelerations at each instant of time. This is referred to as the inverse kinematics problem.

The two reference (nominal) trajectories considered for simulation purposes are discussed below.

TRAJECTORY 1 : It is assumed that the end-effector travels in a semi-circular path with radius r. The center of the circular path is at a distance 'a' from the base of the model robot. This trajectory is shown in fig (2.2). Let the rotation of the tip be in counter-clockwise direction.

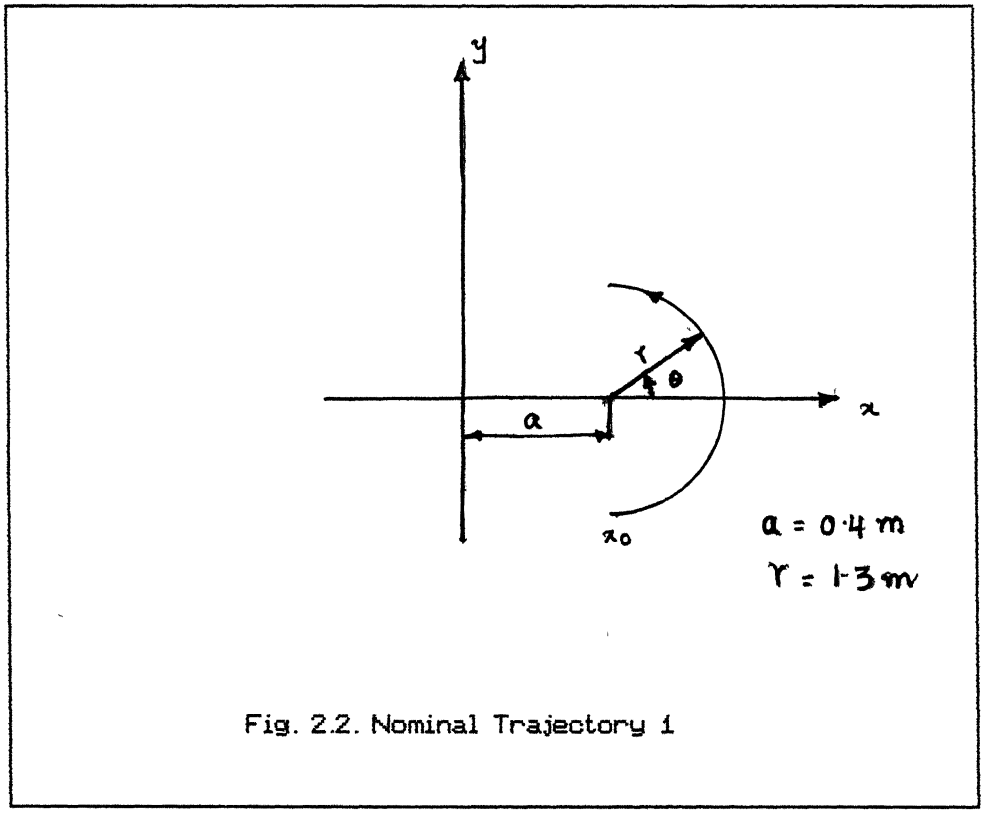


Fig. 2.2. Nominal Trajectory 1

Then the equation of the circle is given by,

$$(x-a)^2 + y^2 = r^2 \qquad (2.5)$$

Considering the angle of rotation to be θ radians, the cartesian points can be determined from the following relationship.

$$x = a + r\cos(\theta)$$

$$y = r\sin(\theta) \qquad (2.6)$$

The trajectory is to be followed in the given fashion.

i) The robot is at rest at a point x_0. Initial joint velocity and acceleration of each joint at this point are zero.

ii) The arm then starts from rest with the end effector moving with a constant acceleration of π rad. /sec.^{2}, until its velocity becomes $\pi/2$ rad /sec.

iii From this point, it moves with constant velocity of π rad / sec. until θ becomes $\pi/8$ rad.

iv) Then it decelerates with constant deceleration of π rad./ sec^2 until θ becomes $\pi/2$ rad. At this point the acceleration and velocity are made zero such that the arm is at rest. The whole process has to be finished in 2 sec.

TRAJECTORY 2: For trajectory 2 it is assumed that the end effector travels in an elliptical path with radii r_1 and r_2. The center of the elliptical path is at a distance 'a' from the base of the model robot. Fig (2.3) shows the end effector trajectory. Let the rotation of the tip be in counter-clockwise direction. Assuming the angle of rotation to be θ radians, the cartesian points can be determined from the following relationship.

$$x = a + r_1\cos(\theta)$$

$$y = r_2\sin(\theta) \qquad (2.7)$$

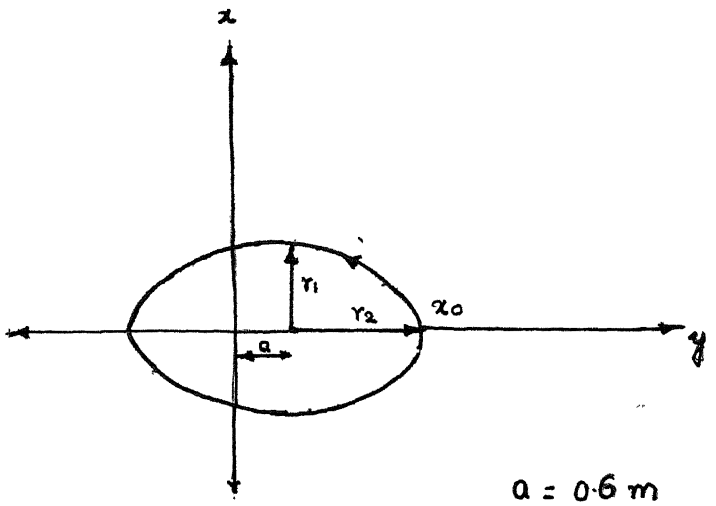


Fig. 2.3 Nominal Trajectory 2

The trajectory is to be followed in the given fashion.

i) The robot is at rest at a point x_0. Initial joint velocity and acceleration of each joint at this point are zero.

ii) The arm then starts from rest with the end effector moving with a constant acceleration of $4\pi/3$ rad./sec.^{2}, until its velocity becomes $2\pi/3$ rad. /sec.

iii) From this point, it moves with constant velocity of $2\pi/3$ rad/ sec. until θ becomes $3\pi/2$ rad.

iv) Then it decelerates with constant deceleration of $4\pi/3$ rad./sec^2 until θ becomes 2π rad. At this point the acceleration and velocity are made zero such that the arm is at rest. The whole process has to be finished in 3 sec.

After generating the two trajectories the next step is to calculate the manipulator joint positions, velocities and accelerations from this information. For constant acceleration or deceleration the following relation holds,

$$d^2\theta / dt^2 = C \qquad (2.8)$$

Where C is a constant whose magnitude is positive if it is acceleration and negative if it is deceleration.

Similarly for constant velocity region of trajectory the following relation holds.

$$d\theta / dt = C \qquad (2.9)$$

The solution of the differential eqns. (2.8) and (2.9) can be obtained by using standard numerical integration methods such as Runge-Kutta etc. For the present purpose, Trapezoidal method is used to solve for θ at every discrete instant k. The values of $\theta(k)\ \forall\ k$ thus computed are stored and subsequently used for finding the reference trajectory using inverse kinematic relationships given below.

$$\cos(q_2) = (x^2 + y^2 - l_1^2 - l_2^2) / 2 l_1 l_2 \qquad (2.10)$$

$$q_1 = \tan^{-1}(y/x) - \tan^{-1}\{[l_2 \sin(q_2)] / [l_1 + l_2 \cos(q_2)]\} \qquad (2.11)$$

The vector $\underline{q}(k)$, containing the joint positions, thus obtained is differentiated to get $\dot{\underline{q}}(k)$ and $\ddot{\underline{q}}(k)$.

The overall algorithm for finding joint trajectory for the circular and elliptical end effector trajectories can be found in [28].

Fig (2.4) and Fig (2.5) show the two nominal joint trajectories ie., the joint position and velocities for the semi-circular trajectory and elliptical trajectory, respectively.

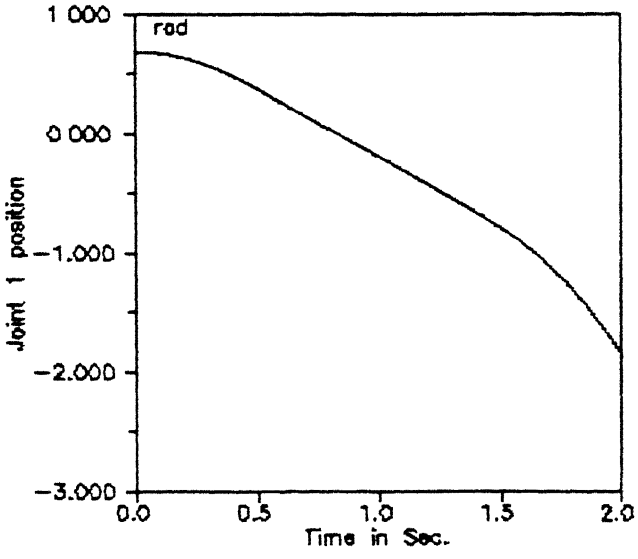
2.3 State Space - Formulation :

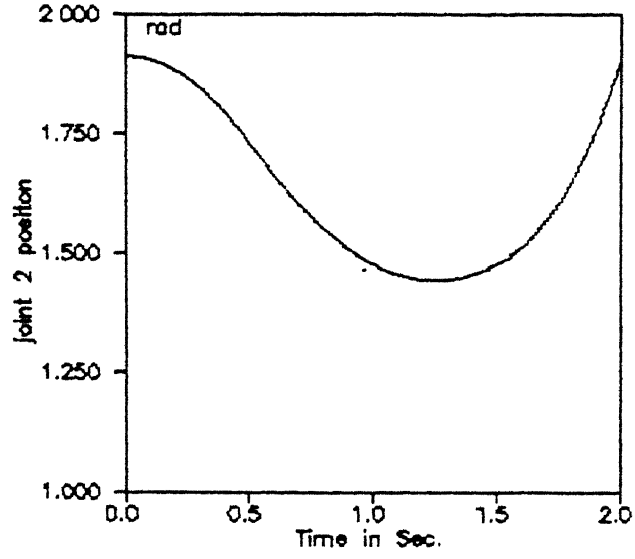
The manipulator dynamic equation (2.1) can be rewritten in order to facilitate the state-space formulation as

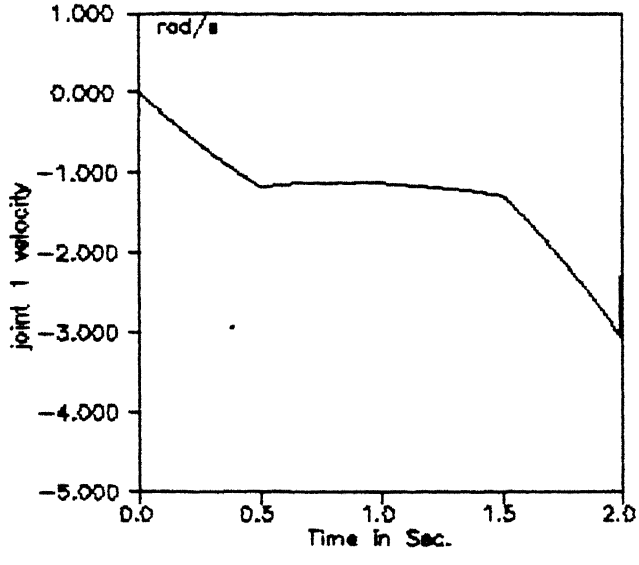
$$\ddot{\underline{q}} = D^{-1}(\underline{q})\{\underline{u} - H(\underline{q}, \dot{\underline{q}}) - g(\underline{q})\} \qquad (2.12)$$

As D is a positive definite matrix representing manipulator inertia, $D^{-1}(\underline{q})$ always exists.

For the trajectory control of a robot manipulator the joint position and velocity vectors are usually feedback through a feedback mechanism. Thus the







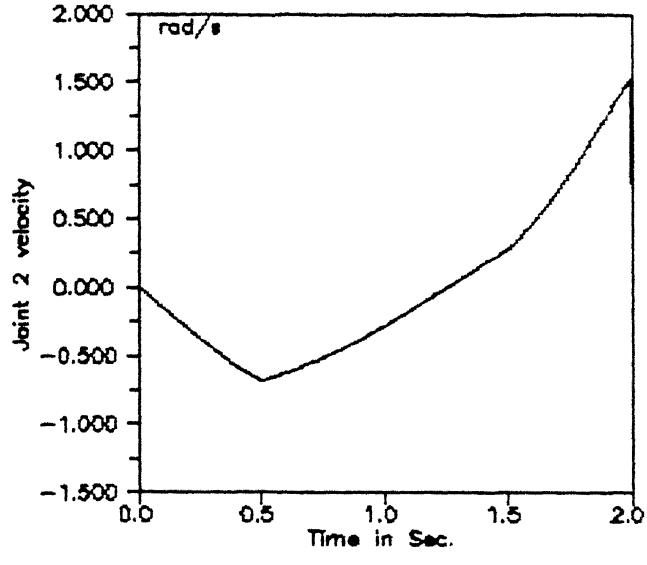
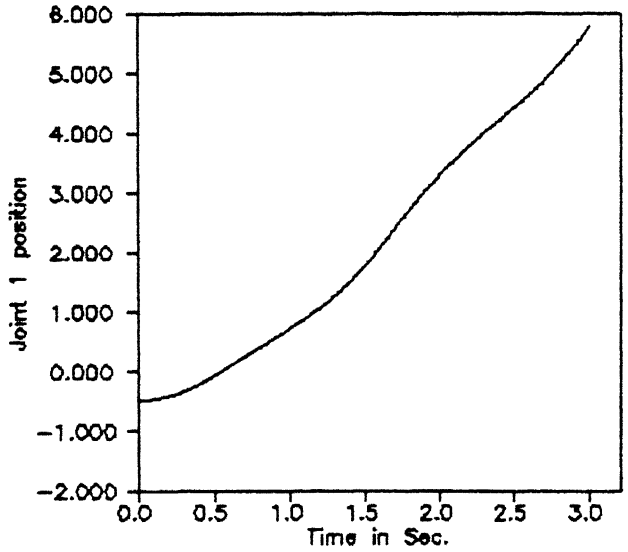
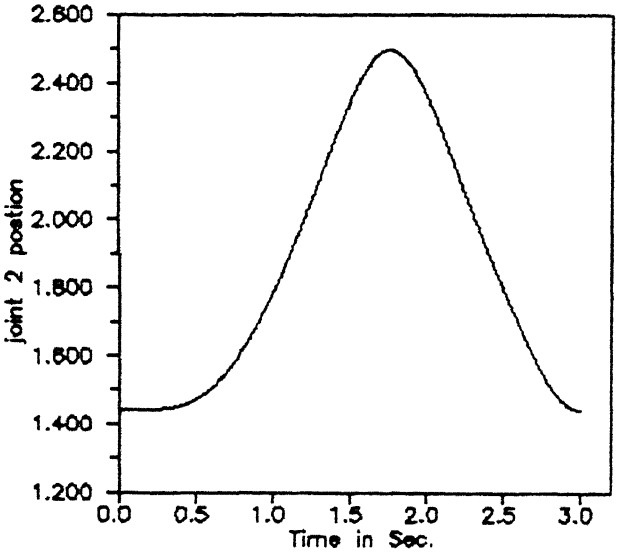
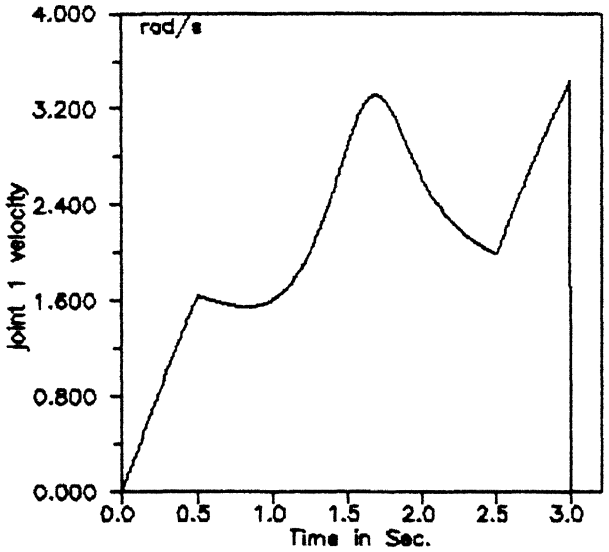


Fig. 2.4 Nominal Trajectory 1







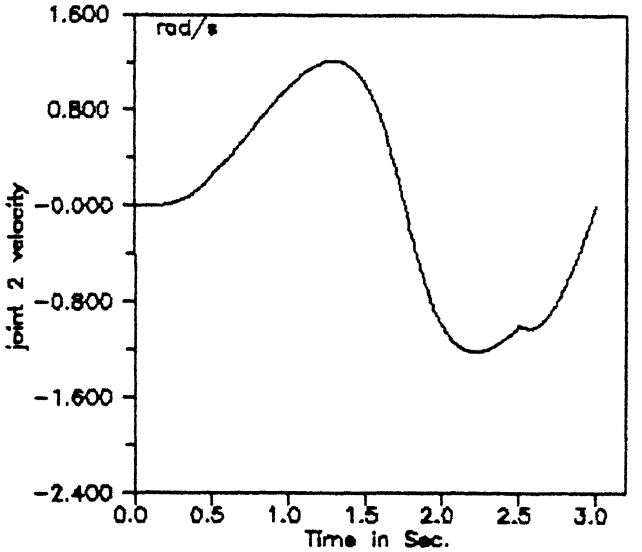


Fig. 2.5 Nominal Trajectory 2

most obvious choice for as state vector is $\underline{x}^T = [\ \underline{q}^T(t)\ \ \dot{\underline{q}}^T(t)\ \ ] = [\ \underline{x}^T(t)\ \ \dot{\underline{x}}^T(t)\ ]$ where $\underline{x}(t) \in \Re^{2n}$. Eq. (2.1) then can be rewritten in terms of the state vector as,

$$\begin{bmatrix} \dot{\underline{q}}(t) \\ \ddot{\underline{q}}(t) \end{bmatrix} = \begin{bmatrix} \dot{\underline{q}}(t) \\ D^{-1}(\underline{q})\ (\ \underline{u} - H(\underline{q}, \dot{\underline{q}}) - g(\underline{q})\ ) \end{bmatrix} \qquad (2.13)$$

which gives the state-space representation of the form

$$\dot{\underline{x}}(t) = \underline{f}(\underline{x}, \underline{u}(t), t)\ . \qquad (2.14)$$

Eqns. (2.13) and (2.14) express the nonlinear manipulator dynamics in terms of state-space representation.

From this state-space representation a suitable controller which will stabilize the closed loop system can be designed by using any of the standard control techniques. But almost all the control schemes assume perfect state information for feedback, i.e., the measurements of the joint positions and velocities are noise free. In a real life situation, the measurements are often corrupted by noise. Thus for good tracking, state estimation technique is necessary to disambiguate the feedback measurements. The following section deals with this problem.

2.4 State Estimation Problem :

The problem is now to estimate the state of a non-linear time varying system through discrete and noisy observations of the system state. Many authors have worked on the development of state estimators for non-linear systems. A survey of the currently available recursive state estimation schemes applicable to a broad class of non-linear systems can be found in [18]. For the present purpose, an Extended Kalman Filter(EKF) and a Constant Gain Extended Kalman Filter (CGEKF) are studied.

2.4 .1 Extended Kalman Filter :

The Extended Kalman Filter, which is used for nonlinear filtering is a natural extension of the Kalman Filter for linear systems. The signal model of the robot manipulator is thus given in terms of eq. (2.14) as,

$$\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{u}, t) + \underline{\omega}(t) \qquad (2.15)$$

$$\underline{y}(k) = \underline{h}(\underline{x}(k), \underline{u}(k), k) + \underline{\nu}(k) \qquad (2.16)$$

Eq. (2.15) is similar to eq. (2.14) except for an additional term $\underline{\omega}(t)$, which represents the process noise .

It is assumed that the output measurements are available at discrete instants Eq. (2.16) represents the output equation, where $\underline{\nu}(k)$ is a vector containing measurement noises, and $\underline{h} \in \Re^{n \times n}$ is a non-linear function. It is assumed that $\underline{\nu}(k) \sim (0, R)$, is white, zero mean and uncorrelated such that $E[\underline{\nu}(k)\underline{\nu}^T(j)] = R\,\delta(k,j)$, where $\delta(k, j)$ is Kronecker delta function defined by

$$\delta(k, j) = 1 \text{ for } k=j$$

$$\delta(k, j) = 0 \text{ for } k \neq j \qquad (2.17)$$

Theoretically the process noise $\underline{w}(t)$ should be absent if the chosen state model accurately represents the robot dynamics. But it is introduced in the state model for the filter tuning purpose. It is assumed that the process noise $\underline{\omega}(t) \sim (0, Q)$ is also white zero mean Gaussian noise and is uncorrelated with $\underline{\nu}(k)$. To solve the state equations (2.15) and (2.16) an initial state $\underline{x}(0)$ has to be defined. However, as the signal model is stochastic, this initial state is also defined as a Gaussian Vector and with added assumption that $E[\underline{x}^T(0)\,\underline{w}(0)] = 0$, $\underline{x}(0) \sim (\bar{\underline{x}}(0), P(0))$, where $Q \in \Re^{2n \times 2n}$ and $P(0) \in \Re^{2n \times 2n}$ are the covariance matrices of $\underline{\omega}(t)$ and $\underline{x}(0)$; $\bar{\underline{x}}(0)$ is $E(\underline{x}(0))$.

Eqns. (2.15) and (2.13) can be rewritten to obtain

$$\begin{bmatrix} \dot{\underline{q}} \\ \ddot{\underline{q}} \end{bmatrix} = \begin{bmatrix} \dot{\underline{q}} \\ D^{-1}(\underline{q})\,(\underline{u} - H(\dot{\underline{q}}, \underline{q}) + g(\underline{q})) \end{bmatrix} + \underline{\omega}(t) \qquad (2.18)$$

It is to be noted that in the signal model, the state equation is given in continuous time while the measurements are taken at discrete instatants. Also note that the measurements are taken for position and velocity of each joint. Then the measurement equation becomes

$$\underline{y}(k) = \begin{bmatrix} I_n & 0 \\ 0 & I_n \end{bmatrix} \begin{bmatrix} \underline{q}(k) \\ \dot{\underline{q}}(k) \end{bmatrix} + \underline{\nu}(k) \qquad (2.19)$$

The step by step algorithm for continuous time EKF with discrete measurements is given below [19].

Algorithm :

Input data : $\bar{\underline{X}}(0)$, $P(0)$ *and* $Q(t)$.

STEP 1 : Measurement update at each instant K is computed as,

Kalman Gain : $$L(K) = P(k/k-1)\,[\,P(k/k-1) + R\,]^{-1} \qquad (2.20)$$

Error covariance : $$P(k/k) = [\,I - L(k)\,]\,P(k/k-1] \qquad (2.21)$$

Estimate : $$\hat{\underline{x}} = \hat{\underline{x}}(k/k-1) + L(k)\,[\,\underline{y}(k) - \hat{\underline{x}}(k/k-1)\,] \qquad (2.22)$$

STEP 2 : Find $F(\hat{\underline{x}},t)$ for $\hat{\underline{x}} = \hat{\underline{x}}(k/k)$ that is evaluate $\partial \underline{f}/\partial \hat{\underline{x}}$.

STEP 3 : Time update of equations are given by

Estimate : $$\dot{\hat{\underline{x}}} = \underline{f}(\,\underline{x}\,,\,\underline{u}\,,\,t\,) + \underline{\omega}(t) \qquad (2.23)$$

Error covariance : $$\dot{P} = F(\hat{\underline{x}}\,,\,t)\,P\,(k/k) + P^{T}(k/k)\,F(\hat{\underline{x}}\,,\,t\,) + Q(t) \qquad (2.24)$$

Integrate $\hat{\underline{x}}$ and $\dot{P}$ given in eqns. (2.23) and (2.24) using Runge-Kutta or any other numerical integration routine to find $\hat{\underline{x}}$ (k/k+1) and P(k/k+1).

For good convergence the filter is to be tuned properly. This means a proper selection of the covariance matrix, Q of pseudo-process noise $\underline{\omega}(t)$ added deliberately to the dynamic model. This is done to control the convergence property of the filter and to take care of model inaccuracy arising from various approximations and accumulated round-off errors. Q can be choøsen as positive definite diagonal matrix so that convergence of each joint position and velocity can be controlled individually.

The covariance matrix P(t) propagated by the filter equations should always remain positive-definite. But during computer simulation P(t) may become non positive definite due to accumulated round off errors, which may result in the filter divergence. In case of severe divergence a square-root algorithm or an U-D algorithm might be required. However, for simpler cases by choosing proper value of Q this problem might be alleviated.

2.4.2 Constant Gain Extended Kalman Filter :

The EKF algorithm discussed in the previous section is computationally intensive and it significantly limits the scope of applications. The major portion of the computational burden results from the the calculations associated with the propagating Error Covariance Matrix P(k), which in turn is used for real time updating of the gain matrix L(k) on the filter residuals. When one considers the gross nature of approximations that are routinely made in modeling the stochastic disturbances effecting a system and to a lesser extent in modeling the interplay between the disturbances and system non-linearities, it seems somewhat surprising that so much computational effort should be devoted to careful propagation of the model's error covariane matrix. Based on these observations, M.S. Safonov and M. Athans [20] developed a Constant Gain Extended Kalman Filter in which the gain matrix acting on the residuals is assumed to be constant. Interestingly, the CGEKF has all the intrinsic robustness to modeling errors as EKF and it drastically reduces the Computational burden.

The Constant Gain Extended Kalman Filter is the filter given by equation (2.22) in which the gain matrix L(k) is a constant given by

$$L = P [P + R]^{-1}$$

here P is the solution of the Ricatti equation (2.25)

$$F(\hat{\underline{x}}, t) P + P^{T} F(\hat{\underline{x}}, t) + Q(t) = 0$$

Algorithm :

STEP 1 : Estimate $\hat{\underline{x}}(k/k) = \hat{\underline{x}}(k/k-1) + L\,[\,\underline{y}(k) - \hat{\underline{y}}\,(k/k-1)\,]$ ' ' (2.26)

STEP 2 : Time update of equation (2.25) is

$$\dot{\hat{\underline{x}}}(t) = \underline{f}(\,\underline{x}\,,\underline{u}\,,t\,) + \underline{\omega}(t)$$

INPUT : $\hat{\underline{x}}(0)$ and L .

The matrix L can be obtained from eq. (2.25) by choosing a proper value of the pseudo process noise covariance matrix **Q**.

2.5 Linear Controller :

The choice of control law may depend upon many features including

i) is the process open loop unstable ?

ii) is the measured output corrupted by noise?

iii) are there are any constraints on available computing power?

iv) is excessive actuator movement undesirable ?

Therefore, the choice of control law is highly application dependent. Ultimately, a trade off must be made between improved quality of control and controller simplicity.

In the present work, a linear quadratic controller is used for the robot control This as the name implies, can only be used on a linear model However, as discussed earlier, the robot model is highly nonlinear and hence has to be linearized. This linearization is done around a nominal trajectory to obtain a perturbation model. This is discussed below.

2.5.1 Linear Perturbation Model :

A linearized perturbation model can be obtained by expanding eq. (2.14) about the nominal trajectory by using Taylor's series and neglecting the higher order terms. This is given by,

$$\delta\dot{\underline{x}} = \nabla_x f\big|_d \, \delta\underline{x}(t) + \nabla_u f\big|_d \, \delta\underline{u}(t)$$

$$= A(t)\,\delta\underline{x}(t) + B(t)\,\delta u(t) \qquad (2.26)$$

Where $A(t) = \nabla_x f\big|_d$ and $B(t) = \nabla_u f\big|_d$ are the jacobian matrices of $f(\underline{x}(t), \underline{u}(t))$ evaluated at $\underline{x}_d(t)$ (the desired state vector at time instant t) and $\underline{u}_d(t)$ (the desired torque vector at time instant t) respectively and

$$\delta\underline{x}(t) = \underline{x}(t) - \underline{x}_d(t) \qquad (2.27)$$

$$\delta\underline{u}(t) = \underline{u}(t) - \underline{u}_d(t) \qquad (2.28)$$

Eq. (2.26) represents a second order linear time varying system. The system parameters A(t) and B(t) vary very slowly with time and depend upon instantaneous manipulator position and velocity . But if the control update speed is increased, then the system can be viewed as a linear time invariant system so that the argument t can be dropped. The nominal torque vector $\underline{u}_d(t)$ can be calculated from an inverse dynamics algorithm like computationally efficient Newton-Euler formulation. The state vector $\underline{x}(t)$, comprising of the joint pastiness and velocities can be assumed to be available for measurements.

The computation of the system matrices A and B poses a serious problem. For a simple two link manipulator, some of the elements of A and B contain over 50 components. So this increases the computational burden which is unavoidable. The next step is to determine a linear quadratic control law based on this perturbation model.

2.5.2 Linear Quadratic controller :

As a result of this formulation, the manipulator control problem is reduced to determining $\delta u(t)$, which drives $\delta x(t)$ to zero at all times along the nominal trajectory. The overall controlled system is thus characterized by a feedforward component and a feedback component Given the planned trajectory set points ($q_d(t)$, $\dot{q}_d(t)$, $\ddot{q}_d(t)$), the feedforward torque computes the corresponding nominal torques $u_d(t)$ from N-E equations of motion. The feedback component computes the corresponding perturbation torques $\delta \underline{u}(t)$ which provides control effort to compensate for small deviations from the nominal trajectory. The main advantages of this formulation are two fold. First, it reduces the non-linear control problem to a linear control problem about a nominal trajectory; second, the computations of the nominal and perturbation torques can be performed separately and simultaneously. After obtaining the linearized perturbation equations the next step is to formulate the control law, which is based on the following performance index.

$$J = \int_0^\infty (\delta \underline{x}^T(t)\ \Gamma\ \delta \underline{x}\ (t) + \delta \underline{u}^T(t)\ \Theta\ \ \delta \underline{u}(t)\ )\ dt \qquad (2.29)$$

where $\Gamma = \Gamma^T \geq 0$ and $\Theta = \Theta^T > 0$

The optimal control law is given by

$$\delta \underline{u}(t) = -\ \Theta^{-1}\ B^T \Upsilon\ \delta \underline{x} \qquad (2.30)$$

where $\Upsilon = \Upsilon^T > 0$ is the solution of the algebraic Ricatti equation,

$$A^T \Upsilon + \Upsilon\ A\ + \Gamma - \Upsilon B\ \Theta^{-1} B^T \Upsilon = 0 \qquad (2.31)$$

The above formulation is based on the assumption that the state measurements are noise free. In the case of noisy measurements modified the performance index becomes

$$J = E\left\{ \int_0^{\infty} (\delta \underline{x}^T(t)\ \Gamma\ \delta \underline{x}(t) + \delta \underline{u}^T(t)\ \Theta\ \delta \underline{u}(t))\, dt \right\} \qquad (2.32)$$

and the control law is given by

$$\delta \underline{u}(t) = -\Theta^{-1} B^T \Upsilon\ \delta \hat{\underline{x}}(t) \qquad (2.33)$$

where $\delta \hat{\underline{x}} = \underline{x} - \hat{\underline{x}}$ and $\hat{\underline{x}}$ is the estimated state vector from the noisy measurements of $\hat{\underline{x}}$ obtained from either eq. (2.22) or eq. (2.25)

The above control scheme differs from the other control schemes based on linear time invariant control schemes in the sense that it takes into account all the interactions among the various joints.

The basic control blocks discussed here are used to design a control scheme which will statbilize the closed loop system in presence of model uncertainties and load variations in some optimum sense. This requires some modifications to be made in the state estimation to take into account the model uncertainties in the system dynamics and in the nominal torque calculation to take into account the load variations, which are dealt in the following chapters.

CHAPTER 3
Robust Controller With Nonadaptive Feedforward Compensation

In this Chapter the various modules described in Chapter 2 are integrated with the necessary modifications required for the robot manipulators. The modification is mainly in the state estimator to take into account the fact that a good representation of ths system dynamics, which is required for the filter convergence is not available due to imprecisely known dynamic parameters. The performance of the control law based on LQ theory in conjunction with the modified estimator algorithm is analyzed by implementing the control scheme on the simulated robot. The simulation results are also presented.

As mentioned in Chapter 2 the control scheme consists of a feedforward term to compensate for the nonlinearities and feedback term based on LQ theory to compensate the parameter perturbations and loading effects. The basic block diagram of the control scheme is shown in fig (3.1). The noisy measurements of joint positions and velocities are estimated by the modified EKF/CGEKF. The feedforward component of the control torque can be calculated from the Newton-Euler formulation and the feedback term based on LQ optimization principle is calculated from the linearized perturbation equations. It can be observed that the feedback and feedforward components of the control torque can be calculated simultaneously.

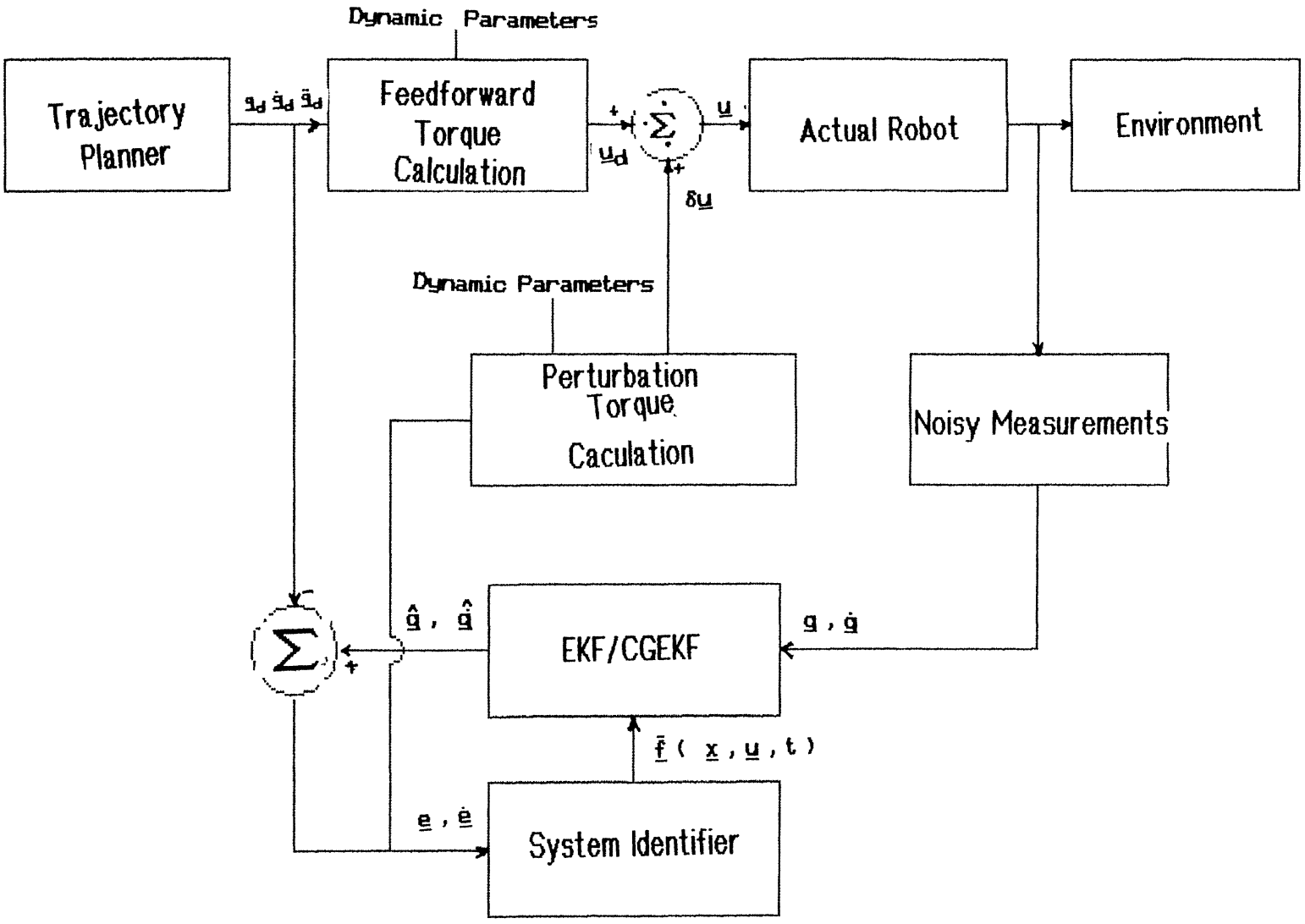


Fig. 3.1 Block Diagram of the Control Scheme

26

3.1 State Estimation with EKF :

Consider the system given below,

$$\dot{\underline{x}} = \underline{f}(\underline{x},\underline{u},t) \qquad (3.1)$$

where $\underline{x} \in \Re^{2n}$ is a non-linear continuous function of $\underline{x}$, $\underline{u}$ and t. In the case of the robot manipulators, neither the function $\underline{f}$ is exactly known nor the exact feedback measurements of state $\underline{x}$ are available. The EKF is used to estimate the state vector $\underline{x}$ from the noisy observations. The model chosen for EKF given in eqn. (2.15) is modified as shown below.

$$\dot{\underline{x}} = \hat{\underline{f}}(\underline{x}, \underline{u}, t) + \underline{w}(t) \qquad (3.2)$$

where $\hat{\underline{f}}(\underline{x}, \underline{u}, t)$ is the estimated dynamics model .

In the theory of EKF there is no in built mechanism for dealing with the model inaccuracy in $\hat{\underline{f}}(\underline{x}, \underline{u}, t)$. Usually the torque/force vector is computed from a feedback control mechanism and an improper compensation by the controller may complicate the problem. In various cases this model inaccuracy may cause serious divergence problem when theoretical model behaviour does not agree with the actual behaviour. In mathematical sense, the divergence of the filter occurs when actual error correlation S_k is not bounded as expected from predicted error covariance P_k or in an apparent divergent problem S_k remains within a bound which is greater than predicted by P_k. S_k and P_k are defined as,

$$S_k = E\left\{ [\, \underline{x}_k^{\,actual} - \hat{\underline{x}}(k/k-1)\,]\,[\, \underline{x}_k^{\,actual} - \hat{\underline{x}}(k/k-1)\,]^T \right\}$$

$$P_k = E\left\{ [\, \underline{x}_k - \hat{\underline{x}}(k/k-1)\,]\,[\, \underline{x}_k - \hat{\underline{x}}(k/k-1)\,]^T \right\} \qquad (3.3)$$

where $\underline{x}_k^{\,actual}$ is the plant state output and $\underline{x}_k$ is model state output. $\underline{x}(k/k-1)$ is the ***not updated*** state at $t = t_k$. So the problem is now of two fold,

- To prevent any possible divergence of the filter output.
- To track the actual system parameter by a simple adaptation law.

Finally, both the filter and tracking controller should work together as a

overall controller.

3.2 State Estimation with CGEKFF :

All the above arguments hold good even for the CGEKF since it is essentially an EKF in which the error covariance matrix is not propagated in order to reduce the comutation.

3.3 Estimation of System Parameters :

Let a trajectory tracking error be defined as

$$e = (\underline{x} - \underline{x}_d) \qquad (3.4)$$

$$\dot{\underline{e}} = \underline{f} (\underline{e}, \underline{u}, t) \qquad (3.5)$$

Let the model of eq. (3.2) be redefined as ,

$$\dot{\underline{x}} = \bar{\underline{f}} (\underline{x}, \underline{u}, t) + \underline{w}(t) \qquad (3.6)$$

where $\bar{\underline{f}} (\underline{x}, \underline{u}, t)$ is the nonlinear function which is used.

In order to take into account the errors due to perturbations in the plant parameters this is defined as [15],

$$\bar{\underline{f}} (\underline{x}, \underline{u}, t) = \hat{\underline{f}} (\underline{x}, \underline{u}, t) - \beta \dot{\underline{e}} \qquad (3.7)$$

where β is a positive definite matrix. The nonlinear functional $\hat{f}$ is the approximate dynamic functional available i.e.,

$$\hat{\underline{f}} (\underline{x}, \underline{u}, t) = \hat{D}^{-1}(q) [\underline{u} - \hat{H}(q,\dot{q}) - \hat{g}(q)] \qquad (3.9)$$

where

$$\hat{D} = D + \Delta D$$

$$\hat{H} = H + \Delta H \qquad (3.10)$$

$$\hat{g} = g + \Delta g$$

ΔD, ΔH and Δg are the uncertainties associated with D, H and g matrices, respectively.

This formulation can be regarded as a joint state and parameter estimation problem in which the system parameters are estimated from the tracking error

information. For proper convergence of the control law, it is important that β be tuned properly. β can be partitioned as [19]

$$\beta = \begin{bmatrix} \beta_1 & 0 \\ 0 & \beta_2 \end{bmatrix} \qquad (3.11)$$

where $\beta_1, \beta_2 \in R^n$ It can be observed that β_1 is associated with the positional components of the state vector and β_2 with the velocity components of the state vector. As the system dynamics has no direct effect on the position vector β_1 can be assumed to be diagonal matrix with diagonal elements nearly equal to one.

The uncertainty matrices can be modeled as norm bounded perturbations, i.e, the norms $\|\Delta D\|$, $\|\Delta H\|$ and $\|\Delta G\|$ are bound by some upper limits. β_2 then can be defined approximately in terms of these uncertainties as

$$\beta_2 = \left[I - D^{-1}(\Delta D^{-1} + D^{-1})^{-1}\right]\left[\ \underline{u} - \hat{H}(\underline{q}, \dot{\underline{q}}) - \hat{g}(\underline{q})\ \right]$$

Based on the limits on the uncertainties of the dynamic parameters, $\|\beta_2\|$ can now be calculated in a straightforward manner.

3.4 Control Strategy :

It is assumed that the measurement noise statistics are available and the nominal values of all the robot dynamic parameters are known. It should be noted the estimator described above does not play any role directly in the control scheme, but it affects the feedback compensation. The basic steps to be followed are given below.

Algorithm :

Initialization :

Initialize matrices Q, R, and P(0) and initial state $\underline{x}$ (0). Find a suitable β by following the arguments given by eq. (3.11) which produces maximum convergence.

For CGEKF, choose L matrix by trial and error to give maximum convergence.

Choose the Γ and Θ matrices of the controller.

Step 1 :

Tune the system parameter of the model $\dot{\underline{x}} = \bar{\underline{f}} (\underline{x} , \underline{u} , t)$ by

$$\bar{\underline{f}} (\underline{x} , \underline{u}, t) = \hat{\underline{f}} (\underline{x} , \underline{u} , t) - \beta \, \underline{e}$$

Step 2 :

Use the EKF or CGEKF algorithm to find the estimates of position and velocity using the system and measurement model as given here.

$$\dot{\underline{x}} = \bar{\underline{f}} (\underline{x} , \underline{u} , t) + \underline{w} (t)$$

$$\underline{y} (k) = \underline{h}_k \, \underline{x}_k + \underline{v}_k$$

and giving output $[\hat{\underline{x}} (t_k)]^T = [\hat{\underline{q}} (t_k) \, \hat{\underline{q}} (t_k)]$

Step 3 :

Find the nominal torque $\underline{u}_d (t)$ from

$$\underline{u}_d (t) = D (\underline{q}_d , \dot{\underline{q}}_d) \ddot{\underline{q}}_d + H (\underline{q}_d , \dot{\underline{q}}_d) \dot{\underline{q}}_d + g (\underline{q}_d)$$

Step 4 :

Evaluate the jacobian matrices A(t) and B(t) given by eq. (2.26) and solve the Ricatti equation (2.32). The perturbation torque is given by, $\delta \underline{u} (t) = - \Theta^{-1} B^T (t) \Upsilon \, \delta \hat{\underline{x}} (t)$, where Υ is the solution of the Ricatti equation.

Apply the control torque $\underline{u} (t) = \underline{u}_d (t) + \delta \, \underline{u} (t)$.

3.5 Numerical Example :

The above design is used to build a controller for the robot model described in chapter 2. Computer simulation is carried out to find the performance of the control logic when a load of 4 Kg. is attached to the second link The state vector for this model is given by ,

$$\underline{x}^T = [\ q_1\ q_2\ \dot{q}_1\ \dot{q}_2\ ]^T$$

Thus the state equation is,

$$\dot{\underline{x}} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ f_1(\underline{q}, \dot{\underline{q}}, \underline{u}, t) \\ f_2(\underline{q}, \dot{\underline{q}}, \underline{u}, t) \end{bmatrix} + \underline{w}(t) \qquad (3.12)$$

The actual expressions for f_1 and f_2 are given Appendix 2

The jacobian matrices **A(t)** and **B(t)** are given by

$$A(t) = \left[\begin{matrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ \partial f_1/\partial q_1 & \partial f_1/\partial q_2 & \partial f_1/\partial \dot{q}_1 & \partial f_1/\partial \dot{q}_2 \\ \partial f_2/\partial q_1 & \partial f_2/\partial q_2 & \partial f_2/\partial \dot{q}_1 & \partial f_2/\partial \dot{q} \end{matrix} \right] \Bigg|_{(\underline{q}_d(t),\ \dot{\underline{q}}_d(t),\ \underline{u}_d(t))}$$

$$B(t) = \left[\begin{matrix} 0 & 0 \\ 0 & 0 \\ \partial f_1/\partial u_1 & \partial f_1/\partial u_2 \\ \partial f_2/\partial u_1 & \partial f_2/\partial u_2 \end{matrix} \right] \Bigg|_{(\underline{q}_d(t),\ \dot{\underline{q}}_d(t),\ \underline{u}_d(t))}$$

The detailed expressions are given in Appendix B.

It is obvious that the performance of the above controller depends upon the cost-function parameters Γ and Θ . The Γ and Θ matrices can be chosen to be diagonal for independent tracking of the joint positions and velocities. The control weighting parameter Θ determines the relative importance which is to be

placed on the penalization of control by cost-function. It follows that a high value of Θ leads to less control variation and a highly damped output, while a low value of Θ leads to a lower variance of tracking error and smaller offset. It is also possible to choose dynamic weights in the solution of the optimal controller to allow error and control signals to be penalized differently in different ranges. Thus Γ and Θ are given by

$$\Gamma = \text{diag} (\Gamma_{11} , \Gamma_{22}) \quad \text{and} \quad \Theta = \text{diag} (\Theta_{11} , \Theta_{22} , \Theta_{33} , \Theta_{44})$$

For the filter the measurement equation is given by eq. (3.6) As direct sensory measurements of both velocity and position are used, the matrix h_k is assumed to be linear and equal to identity matrix. It is assumed that the measurement noise covariance matrix R is diagonal such that each measurement noise is uncorrelated with others . Then,

$$R = \text{diag} (\alpha^2_{q1}, \alpha^2_{q2}, \alpha^2_{\dot{q}1}, \alpha^2_{\dot{q}2})$$

To select $\hat{\underline{x}}(0)$ a priori knowledge of initial positions and velocities are required. To start the algorithm $\hat{\underline{x}}(0)$ is set to measured value of initial positions and velocities . Next step is to compute P(0) from $E [\| \underline{x} (0) - \hat{\underline{x}} (0) \|^2]$. So from the very choice of $\hat{\underline{x}}(0)$, P(0) becomes P(0) = R. To run the algorithm the jacobian matrix $F(\underline{x} ,t)$ is required which is similar to the A(t) matrix given above. This is given by,

$$F (\underline{x} ,t) = \left. \frac{\partial \underline{f} (\underline{x} , \underline{u} , t)}{\partial \underline{x}} \right|_{\hat{\underline{x}}(k/k)}$$

$$= \left[\begin{array}{cccc} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ \partial f_1/\partial q_1 & \partial f_1/\partial q_2 & \partial f_1/\partial \dot{q}_1 & \partial f_1/\partial \dot{q}_2 \\ \partial f_2/\partial q_1 & \partial f_2/\partial q_2 & \partial f_2/\partial \dot{q}_1 & \partial f_2/\partial \dot{q}_2 \end{array} \right] \Bigg|_{(\hat{\underline{x}}(k/k)}$$

With the above formulation, the the control algorithm is applied to the simulated robot arm. Parametric variations are incorporated into basic dynamic parameters, i.e., the mass of a link, moment of inertia of a link and the center of mass. Table (3.1) gives the list of dynamic parameters used for generating the control logic.

The actual dynamic parameters are already presented in table (2 1)

Table 3.1

link	Mass (Kg)	Length of the link i (m.)	Distance of c.m of link i from i-th coordinate frame(m.)	Inertia of Link i (m)
1	$m_1 = 14$	$l_2 = 1.4$	$lc_1 = 0.45$	$I_1 = 3.75$
2	$m_2 = 8.6$	$l_2 = 0.8$	$lc_2 = 0.175$	$I_2 = 0.5$

The covariance matrix of the pseudo process noise vector $\mu(t)$ is chosen by trial and error method. Q is chosen to be diagonal with positive entries, whose values are adjusted to give the best convergence.

Now, coming to the estimation of noisy feedback measurements, noise variance of the position transducers is assumed to be small where as the noise variance of the velocity transducers is assumed to be comparatively larger to reflect the relative inaccuracy of the velocity sensors due to their low resolution. The matrix L in the case of CGEKF is chosen by trial and error to give maximum covergence. Table (3.2) shows the design parameters of the filter and controller combination.

The simulation runs for the control scheme are performed for the two different state trajectories discussed in Chapter 2 by employing the two state estimation schemes EKF and CGEKF. Table (3.3) shows the maximum tracking errors for the trajectories 1 and 2 with the two state estimators EKF and CGEKF in the control logic and table (3.4) shows the corresponding variances of the tracking errors. The maximum error refers to the maximum deviation from the desired state trajectory. Fig (3.2), fig (3.3) show the tracking errors for the two nominal trajectories with EKF as the state estimator in the control scheme and fig (3.4) and fig (3.5) show the corresponding plots with CGEKF as the state estimator. It can be observed from these figures that the control scheme with EKF outperforms the one with the CGEKF. This is quite expected as, in EKF the error covariance matrix is propagated whereas in the case of CGEKF, it is not. It can be found

that the tracking errors are somewhat large in all the cases, which can be attributed to the fact the actual dynamics of the system is perturbed and the nominal torque obtained form the the model of the system is not the nominal torque required by the manipulator under perturbed or loading conditions. This observation forms the starting point for the control scheme discussed in the next chapter.

Table 3.2

Design parameters of the tracking controller :

Q = diag (1000, 1000, 1000, 1000)

R = diag (1, 1)

Design parameters of the EKF :

R = diag(0.001, 0.001, 0.05, 0.05)

Q = diag(0.0012 ,0.0012, 0.0005, 0.0005)

P(0)= R

β = diag(0.9 , 1.3, 2.1 , 3.1)

Design parameters of the CGEKF :

R = diag(0.001 , 0.001, 0.05 , 0.05)

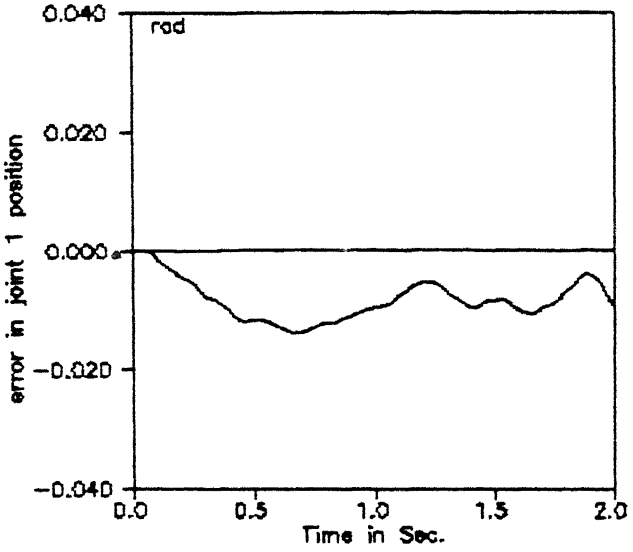
L = diag (0.01475, 0.01475, 0.15, 0.15)

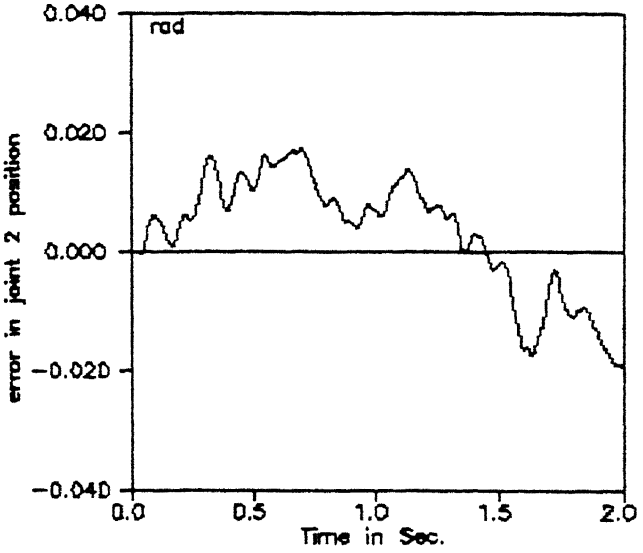
Table 3.3

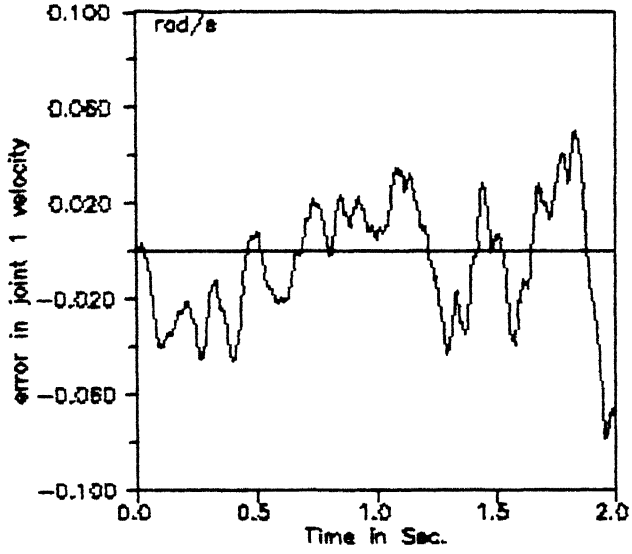
Trajectory	joint	Maximum Tracking Error with EKF		Maximum Tracking Error with CGEKF	
		position (rad)	velocity (rad/s)	position (rad)	velocity (rad/s)
Semi-Circular	1	0.01383	0.07818	0.02134	0.06121
	2	0.01921	0.25876	0.02587	0.30799
Elliptical	1	0.02266	0.07677	0.04545	0.15464
	2	0.01560	0.19864	0.02427	0.41064

Table 3.4

Trajectory	joint	Variance of Tracking Error with EKF		Variance of Tracking Error with CGEKF	
		position (rad)	velocity (rad/s)	position (rad)	velocity (rad/s)
Semi-Circular	1	0.00036	0.00163	0.00176	0.00389
	2	0.00026	0.00321	0.00052	0.00544
Elliptical	1	0.00038	0.00108	0.00559	0.00185
	2	0.00064	0.00870	0.00474	0.00886







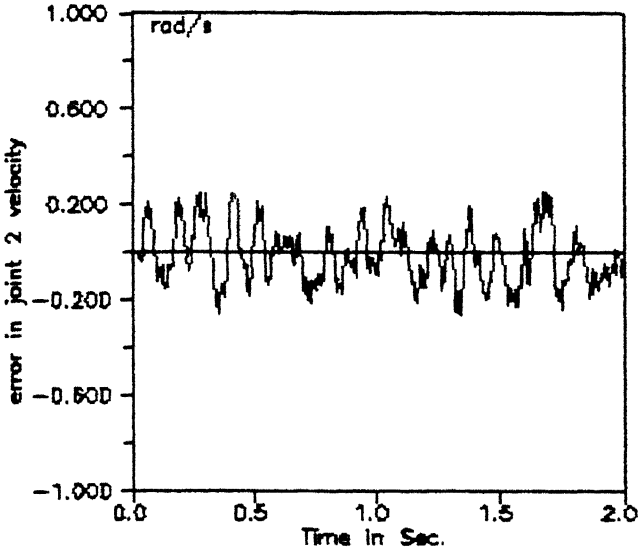
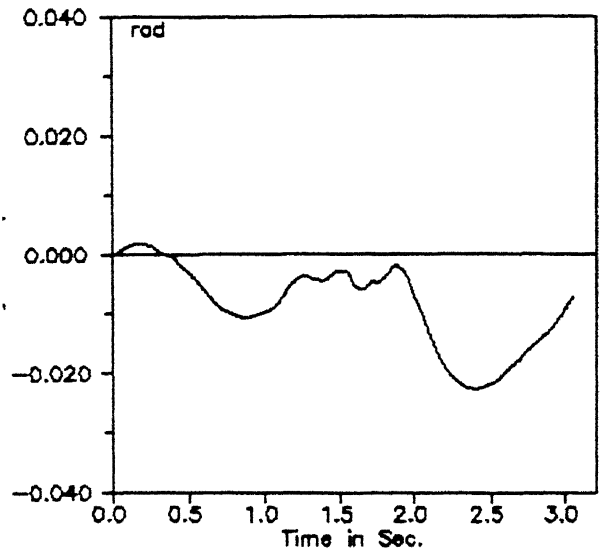
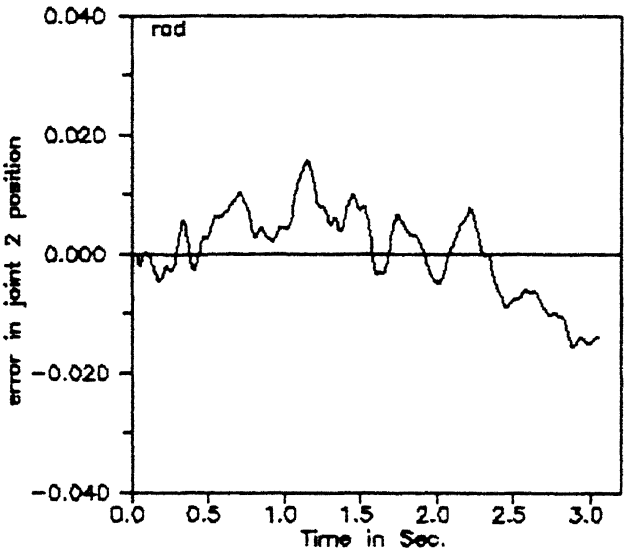
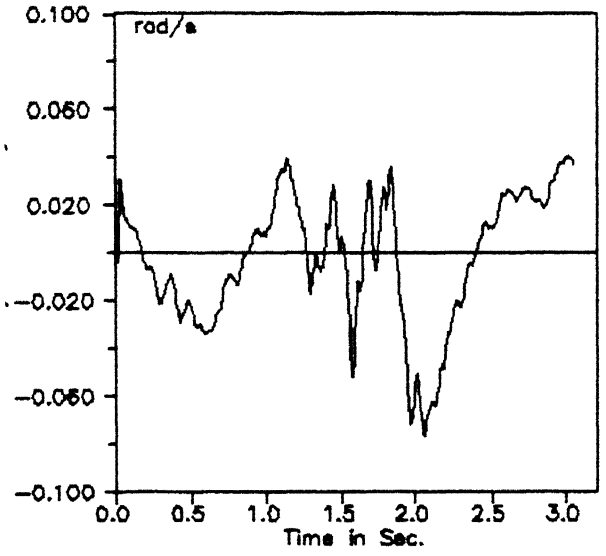


Fig. 3.2 Tracking Error Plot with EKF for Trajectory 1







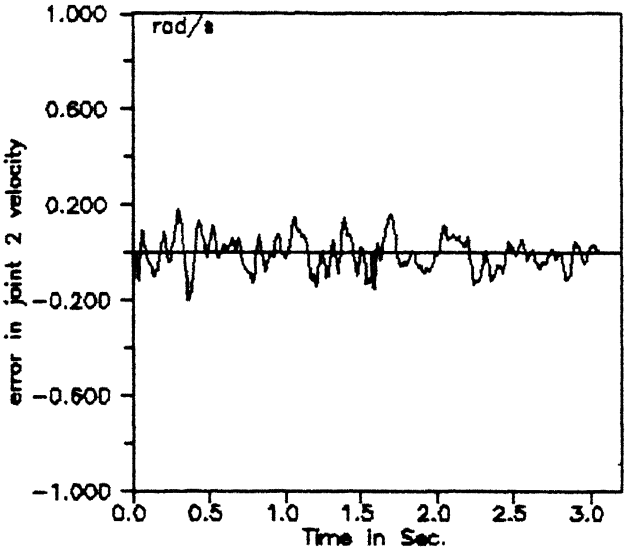
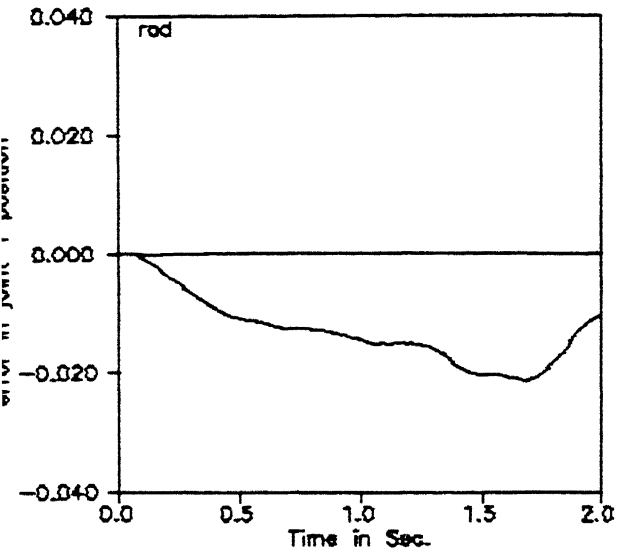
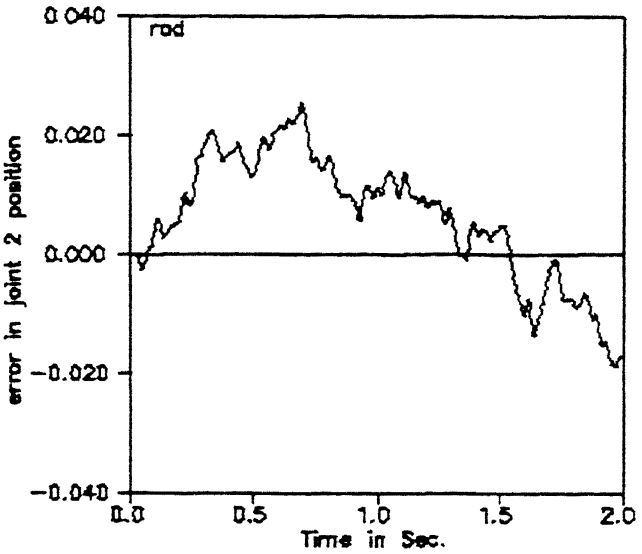
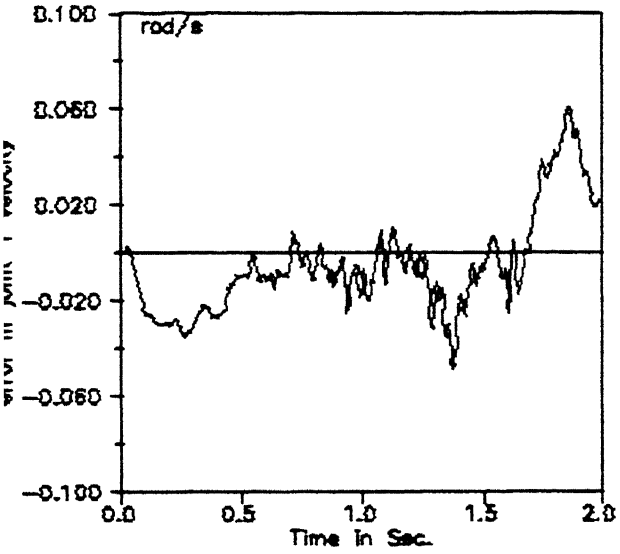


Fig. 3.3 Tracking Error Plot with EKF for Trajectory 2







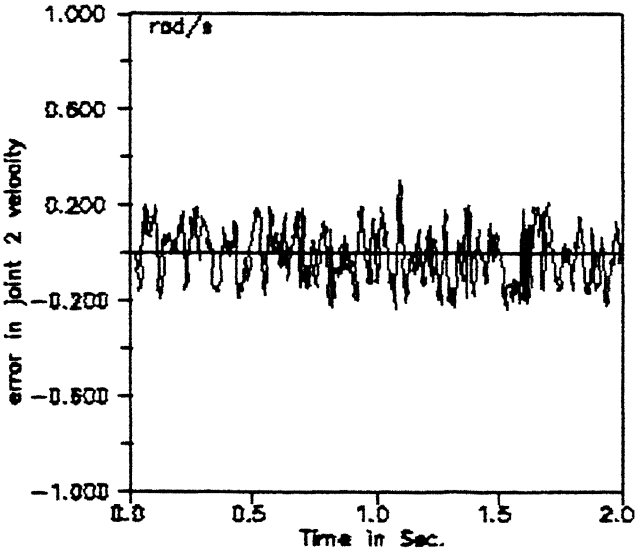
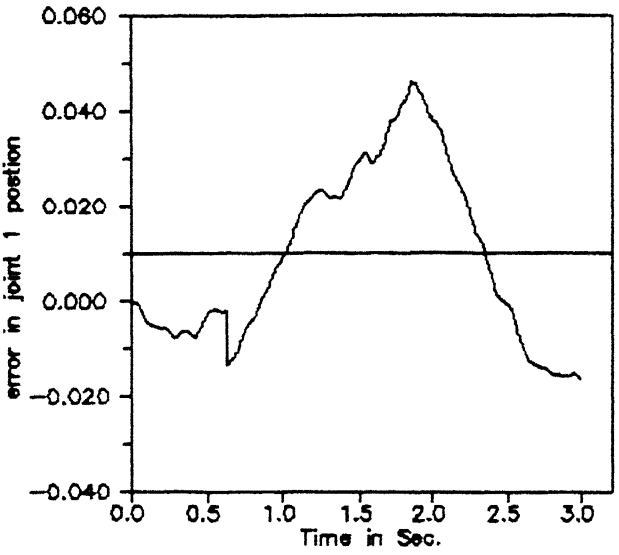
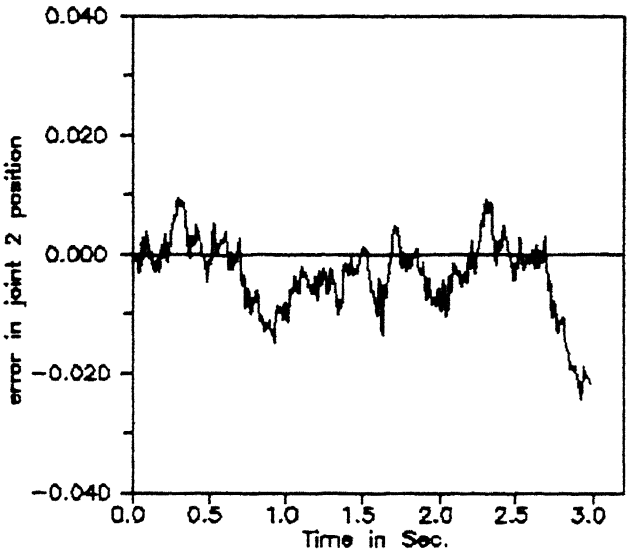
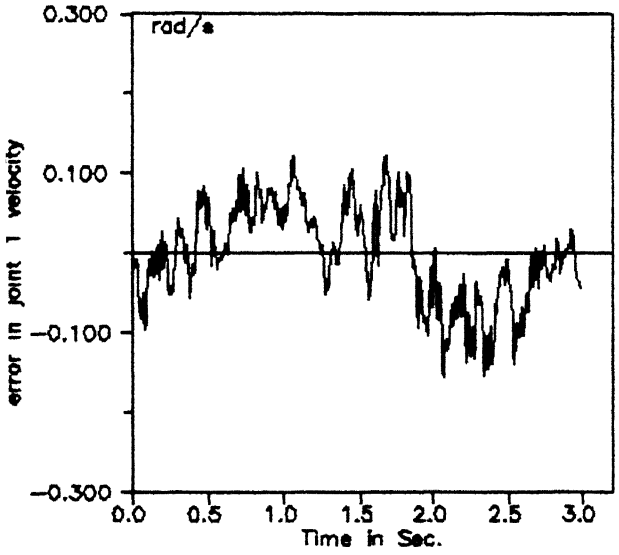


Fig. 3.4 Tracking Error Plot with CGEKF for Trajectory 1







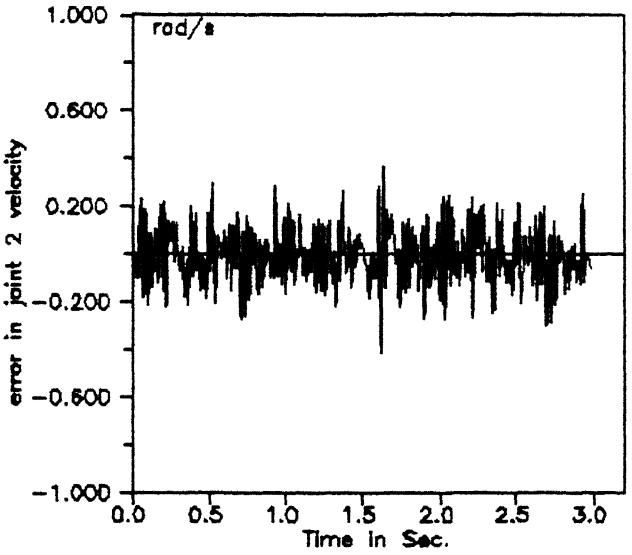


Fig. 3.5 Tracking Error Plot with CGEKF for Trajectory 2

CHAPTER 4

Control Scheme With Adaptive Feedforward Compensation

In this Chapter an adpaptive control scheme for the robot manipulators is suggested. The method presented in this chapter is inspired from the previous control scheme discussed. Instead of using the nominal torque from a model which presumedly is not exact, the nominal torque is calculated from a model based on identified robot dynamics. A recursive identification scheme is used to identify the dynamic parameters. Thus both the loading effects and parameter perturbation effects are captured at once. As in the case of the previous scheme, the closed loop system consists of linearized dynamics obtained from perturbation equations and a LQ type of optimization is used. Finally, the performance the above control scheme in presence of measurement noise is studied by implementing this modified control scheme on the simulated two degree of freedom planar manipulator.

4.1 Dynamic parameter Estimation :

The dynamic model used in computing the nominal torque is given by eq. (2.1) which can be computed recursively by using NE formulation. The nominal torque does not fully compensate for loading or parameter perturbation effects, since it is calculated from the known imperfect information about the dynamic parameters. In principle, the linear controller discussed in previous Chapter would compensate for these disturbances if the perturbations are small and in the vicinity of the nominal values. Below an approach is presented where the nominal torque is computed and compensated based on identified dynamic parameters using recursive

least squares technique

Knowledge of manipulator loads and links (mass, center of mass and moment of inertia) is potentially important for the precise control of movement. Robot manipulators are designed according to precise kinematic specifications, but the link parameters are incidental attributes of design. The NE equations describing the robot dynamics can be recast in such a way to relate linearly the measured torques or forces to the dynamic parameters, which can be estimated by any standard identification technique [20]. In this section one of the identification schemes based on recursive least squares techniques is discussed.

Consider the NE equations (A.1) and (A.2) derived in Appendix A. These equations are clearly nonlinear with respect to the dynamic parameters of each link i. But when they are rewritten by expressing the moment of inertia I_i about the joint origin i instead of about the center of mass C_i, these equations become linear with respect to a set of equivalent dynamic parameters. For example, consider the two link planar manipulator model discussed in Appendix A. In eqns. (A.3), (A.4), (A.5) the terms $(f_{01} - f_{12})$, f_{12}, and $(N_{01} - N_{12})$, N_{12} represent the net force and net torque due to the motion of the individual links 1 and 2 alone, respectively, which can be measured by means of force-torque sensors. Then eqns. (A.1) to (A.5) are rewritten to obtain the following equations,

$$f_{12} = m_2 \dot{v}_{c2} - m_2 g \qquad (4.1)$$

$$f_{01} - f_{12} = m_1 v_{c_1} - m_1 g \qquad (4.2)$$

$$N_{12} = I_2 \dot{w}_2 + r_{1c_2} \times f_{12} \qquad (4.3)$$

$$N_{01} - N_{12} = I_1 \dot{w}_1 + r_{1c_2} \times f_{01} + r_{1c_2} \times f_{12} \qquad (4.4)$$

Using the fact that for the two degree of freedom planar manipulator $N_{01} = u_1$ and $N_{12} = u_2$ the following equations are defined.

$$f_{22} = m_2 \dot{v}_{c2} - m_2 g \qquad (4.5)$$

$$f_{11} = m_1 \dot{v}_{c_1} - m_1 g \qquad (4.7)$$

where f_{11} and f_{22} are the forces at joint 1 and 2 due to the motion of the

individual joints alone.

Let a set of auxilary parameters be defined as

$$I_1' = I_1 + m_1 lc_1^2$$

$$I_2' = I_2 + m_2 lc_2^2$$

$$lc_1' = m_1 lc_1 \qquad (4.8)$$

$$lc_2' = m_2 lc_2$$

Then rewriting the dynamic equations of the two degree of freedom planar robot manipulator can be rewritten using the above relations,

$$u_1 = [\ddot{q}_1] I_1' + [\ddot{q}_1 + \ddot{q}_2] I_2' + [l_1^2 + g l_1 \cos(q_1)] m_2 + lc_2' [-2 l_1 \cos(q_2) \ddot{q}_1 - l_1 \sin(q_2) \dot{q}_2^2 - 2 l_1 \sin(q_2) \dot{q}_1 \dot{q}_2 + +g \cos(q_1 + q_2)] + [g \cos(q_1)] lc_1' \qquad (4.9)$$

$$u_2 = (\ddot{q}_1 + \ddot{q}_2) I_2' + lc_2' [\ddot{q}_1 \cos(q_2) \ddot{q}_1 + 2 \sin(q_2) \dot{q}_1^2 + g \cos(q_1 + q_2)]$$

$$f_{22} = \begin{bmatrix} [-\ddot{q}_1 \sin(q_1) - \dot{q}_1^2 \cos(q_1)] lc_1' \\ [\ddot{q}_1 \cos(q_1) - \dot{q}_1^2] lc_1' - g[m_2] \end{bmatrix}$$

The elements f_{11_i}, $i = 1, 2$ of the vector f_{11} are given by

$$f_{11_1} = -[l_1 \cos(q_1) \dot{q}_1^2 + l_1 \sin(q_1) \ddot{q}_1] m_2 - [\cos(q_1+q_2)(\dot{q}_1 + \dot{q}_2)^2 + \sin(q_1+q_2)(\ddot{q}_1 + \ddot{q}_2)] lc2'$$

$$f_{11_2} = -[\sin(q_1) \dot{q}_1^2 - \cos(q_1) \ddot{q}_1 + g] m_2 - [\sin(q_1+q_2)(\dot{q}_1 + \dot{q}_2)^2 - \cos(q_1 + q_2)(\ddot{q}_1 + \ddot{q}_2)] lc_2'$$

The above equations are linear with respect to the selected equivalent auxilary dynamic parameters namely, I_1', I_2', lc_1', lc_2', and the link masses m_2 and m_1. The above equations are combined in a vector matrix form to obtain the following equations.

$$W = Y \Phi \qquad (4.10)$$

where $W = \begin{bmatrix} u_1 \\ f_{11} \\ u_2 \\ f_{22} \end{bmatrix}$, $\Phi = \begin{bmatrix} m_1 \\ lc_1' \\ I_1' \\ m_2 \\ lc_2' \\ I_2' \end{bmatrix}$

and Y is a (6 X 6) matrix whose elements y_{ij} (i = 1...6 , j = 1...6) are given by,

$y_{12} = g \cos(q_1)$

$y_{13} = \ddot{q}_1$

$y_{14} = l_1^2 \ddot{q}_1 + l_1 g \cos (q_1)$

$y_{15} = 2 l_1 \cos (q_2) \ddot{q}_1 + l_1 \cos (q_2) \ddot{q}_2 - l_1 \sin (q_2) \dot{q}_2^2 - 2 l_1 \sin (q_2) \dot{q}_1 \dot{q}_2 + g \cos (q_1 + q_2)$

$y_{16} = \ddot{q}_1 + \ddot{q}_2$

$y_{22} = - [\cos (q_1) \dot{q}_2^2 + \ddot{q}_1 \sin (q_1)]$

$y_{31} = g$

$y_{32} = \cos (q_1) \ddot{q}_1 - \dot{q}_1 \dot{q}_1 \sin (q_1)$

$y_{45} = l_1 \cos (q_2) \ddot{q}_1 + g \cos (q_1 + q_2) + l_1 \sin (q_2) \dot{q}_1^2$

$y_{46} = \ddot{q}_1 + \ddot{q}_2$

$y_{54} = - [l_1 \sin (q_1) \ddot{q}_1 + l_1 \cos (q_1) \dot{q}_1^2]$

$y_{55} = - l_1 \cos (q_1+q_2) (\dot{q}_1 + \dot{q}_2)^2 - \sin (q_1 + q_2) \ddot{q}_2$

$y_{64} = l_1 \cos (q_1) \ddot{q}_1 + g - l_1 \sin (q_1) \dot{q}_1^2$

$y_{65} = \cos (q_1 + q_2) \ddot{q}_2 + \cos (q_1 + q_2) \ddot{q}_1 - \sin (q_1 + q_2) (\dot{q}_1 + \dot{q}_2)^2$

The rest of the elements of y are zeros.

The elements of Y can be calculated at each sampling instant from the measurements of joint angular positions, velocities and accelerations. Generally, the acceleration measurements are derived from the velocity measurements. So the acceleration measurements can be obtained by differentiating the velocity measurements. The elements of W matrix can be derived from the torque/force

sensors So from eq. (4.10) a Recursive least squares scheme can be formulated to identify the Φ vector at each sampling instant, from the measurements of joint positions, velocities and forces/torques. This algorithm is given below.

Algorithm :

Input data : $\hat{\Phi}(0)$, $P(0)$ and λ

Parameter update at each instant k is computed as,

$$\hat{\Phi}(k) = \hat{\Phi}(k-1) + P(k)\, Y^T(k) \, [\, W(k) - Y(k-1)\, \hat{\Phi}(k-1) \,] \qquad (4.11)$$

with

$$P(k) = 1/\lambda \; P(k-1)\, Y^T(k-1) \, [\, \lambda I + Y(k-1)\, P(k-1)\, Y^T(k-1) \,]^{-1} \; Y(k-1)\, P(k-1) \,] \qquad (4.12)$$

where $0 \leq \lambda \leq 1$ is a forgetting factor which provides an exponential weighting of the past data in the estimation algorithm and by which the algorithm allows a drift of parameters.

P is a symmetric positive definite matrix of appropriate dimension.

4.2 Control scheme :

For the purpose of the formation of the control law the state equation representing the robot dynamics, i.e., eq. (2.4) is modified as

$$\dot{\underline{x}} = \hat{f}(\underline{x}, \underline{u}, t) \qquad (4.13)$$

where $\hat{f}$ ( ) is the vector function f ( ) evaluated at the identified dynamic parameters. With the above modification the linearized perturbation equations (2.26) , (2.27) and (2.28) can be modified as

$$\delta\dot{\underline{x}} = \nabla_x \hat{f}\big|_d \, \delta\underline{x}(t) + \nabla_u \hat{f}\big|_d \, \delta u(t) \qquad (4.14)$$

$$= \hat{A}(t)\, \delta\underline{x}(t) + \hat{B}(t)\, \delta u(t)$$

where $\nabla_x \hat{f}\big|_d$ and $\nabla_u \hat{f}\big|_d$ are the jacobian matrices of $\hat{f}(x(t), \underline{u}(t))$ evaluated at $\underline{x}_d(t)$ (the desired state vector at time instant t) and $\hat{\underline{u}}_d(t)$ (the desired torque vector at time instant t) obtained form the NE equations with the identified dynamic parameters. i.e.,

$$\hat{\underline{u}}_d = \hat{D}\,(\,\underline{q}_d\,)\,\ddot{\underline{q}}_d + \hat{H}(\,\underline{q}_d\,,\,\dot{\underline{q}}_d\,)\,\underline{q}_d + \hat{\underline{g}}(\,\underline{q}_d\,) \qquad (4.15)$$

$$\delta\underline{x}(t) = \underline{x}\,(t) - \underline{x}_d\,(t) \qquad (4.16)$$

$$\delta\underline{u}(t) = \underline{u}\,(t) - \hat{\underline{u}}_d\,(t) \qquad (4.17)$$

The optimal control law which minimizes the following performance index

$$J = \int_0^{\infty} (\delta\underline{x}^T(t)\,\Gamma\,\delta\underline{x}\,(t) + \delta\underline{u}^T(t)\,\Theta\;\;\delta\underline{u}(t)\,)\,dt \qquad (4.18)$$

where $\Gamma = \Gamma^T \geq 0$ and $\Theta = \Theta^T > 0$ is given by

$$\underline{u}(t) = -\;\Theta^{-1}\,\hat{B}^T\,\Upsilon\,\delta\underline{x}\,(t) \qquad (4.19)$$

where $\Upsilon = \Upsilon^T > 0$ is the solution of the algebraic Ricatti equation,

$$\hat{A}^T\Upsilon + \Upsilon\,\hat{A} + \Gamma - \Upsilon\hat{B}\,\Theta^{-1}\hat{B}^T\Upsilon = 0 \qquad (4.20)$$

The above formulation is based on the assumption that the state measurements are noise free. In the case of noisy measurements the modified performance index becomes

$$J = E\left\{\int_0^{\infty} (\delta\underline{x}^T(t)\,\Gamma\,\delta\underline{x}\,(t) + \delta\underline{u}^T(t)\,\Theta\;\;\delta\underline{u}(t)\,)\,dt\right\} \qquad (4.21)$$

and the optimal control law which minimizes the above performance index is given by

$$\underline{u}\,(t) = -\;\Theta^{-1}\,\hat{B}^T\,\Upsilon\,\delta\hat{\underline{x}} \qquad (4.22)$$

where $\delta\hat{\underline{x}} = \hat{\underline{x}} - \underline{x}_d$

and $\hat{\underline{x}}$ is the estimated state vector from the noisy measurements of $\underline{x}$ obtained from either eq. (2.15) or eq. (2.16)

With the identifier introduced into the system the modified control scheme is shown in fig (4.1). First the noisy measurements are filtered using either of the two state estimation schemes and the filtered measurements are fed to the identifier. The identified dynamic parameters are used to update the nominal control (feedforward) component using NE equations and simultaneously to generate the perturbation matrices $\hat{A}\,(t)$ and $\hat{B}(t)$.

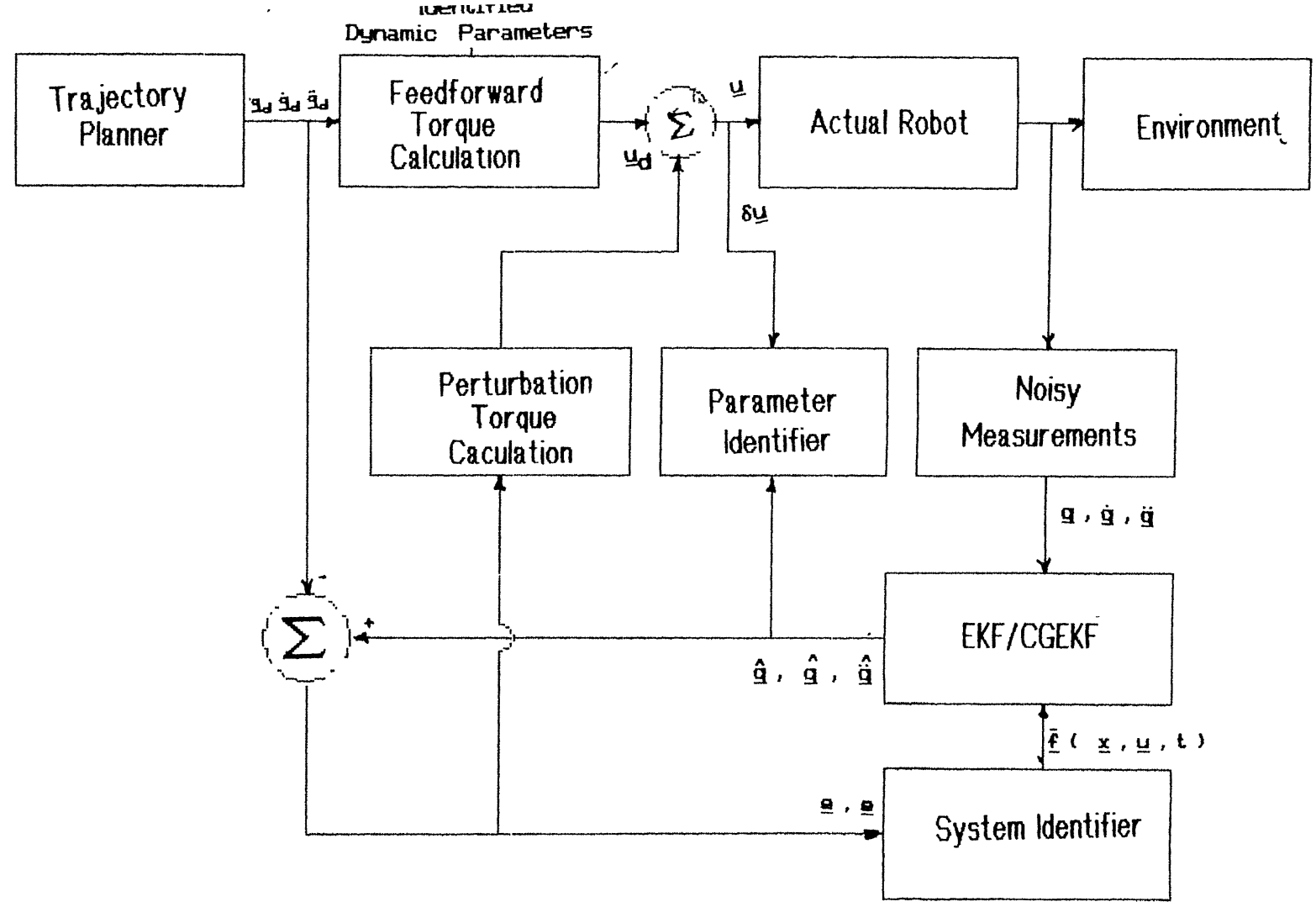


Fig. 4.1 Block Diagram of the Control Scheme With Adaptive Feed-forward Compensation

A step by step algorithm for the total control scheme is given below.

Algorithm :

Initialization :

Intialize matrices Q, R, and $P(0)$ and initial state $\underline{x}(0)$. Find suitable β by following the guidelines given in eq. (3.11).

For CGEKF, choose L matrix by trial and error to give maximum convergence.

Initialize $\Phi(0)$, $P(0)$ of the identifier by using the a priori information about the dynamic parameters of the manipulator.

Initialize the Γ and Θ matrices of the controller.

Step1 :

Tune the system parameter of the model $\dot{\underline{x}} = \bar{\underline{f}}(\underline{x}, \underline{\tau}, t)$ by

$\bar{\underline{f}}(\underline{x}, \underline{u}, t) = \hat{\underline{f}}(\underline{x}, \underline{u}, t) - \beta \dot{\underline{e}}$

Step 2 :

Use the EKF or CGEKF algorithm to find the estimates of position and velocity using the system and measurement model as given here.

$\dot{\underline{x}} = \bar{\underline{f}}(\underline{x}, \underline{u}, t) + \underline{w}(t)$

$\underline{y}(k) = h_k \underline{x}_k + \underline{v}_k$

and giving output $[\hat{\underline{x}}(t_k)]^T = [\hat{\underline{q}}(t_k)\ \hat{\dot{\underline{q}}}(t_k)]$

Differentiate the estimated velocity vector to obtain the acceleration vector $\hat{\ddot{\underline{q}}}$.

Step 3 :

Run the identifier to find the estimates of dynamic parameters using the torque /force measurements, the filtered measurements $\hat{\underline{q}}$, $\hat{\dot{\underline{q}}}$ and the derived acceleration measurements $\hat{\ddot{\underline{q}}}$ Find the nominal torque from the following relation

$\hat{\underline{u}}_d(t) = \hat{D}(\underline{q}_d, \dot{\underline{q}}_d)\ddot{\underline{q}}_d + \hat{H}(\underline{q}_d, \dot{\underline{q}}_d)\dot{\underline{q}}_d + \hat{\underline{g}}(\underline{q}_d)$

where $\underline{q}_d$, $\dot{\underline{q}}_d$, $\ddot{\underline{q}}_d$ are the desired angular position , velocity

and acceleration of the joints respectively and $\hat{D}$, $\hat{H}$ and $\hat{g}$ are as defined in eq (2.1)

Step 4 :

Evaluate the jacobian matrices $\hat{A}(t)$ and $\hat{B}(t)$ given by eq. (2.26) and solve the Ricatti equation (2.32) . The perturbation torque is given by,

$\delta\underline{u}(t) = -\Theta^{-1}\hat{B}^{T}(t)\,\Upsilon\delta\hat{\underline{x}}(t)$, where Υ is the solution of the Ricatti equation. The control input to the system is then

$\underline{u}(t) = \underline{u}_{d}(t) + \delta\,\underline{u}(t)$.

4.3 Numerical Examples :

To check the performance of the identification scheme and the control scheme discussed in this Chpater experiments are conducted on the simulated two degree of freedom planar manipulator which are discussed below.

4.3.1 Simulation Experiment for Identification Scheme :

The identification scheme discussed is implemented on the simulated two degree of freedom manipulator. To make the simulation realistic the measurements are corrupted with noise. It is found that the algorithm is very robust and has very good convergence property. It is observed that the above algorithm has high sensitivity to the noise in the acceleration measurements. Table (4.1) shows the simulation parameters Fig . (4.6) shows the simulation results.

4.3.2 : Simulation Experiment for the Overall Control Scheme :

The performance of the above control algorithm is studied by conducting tests on the simulated two degree of freedom planar manipulator. The simulation tests are carried out separately with the two state estimators namely, EKF and CGEKF. The actual robot effect is simulated by using the actual values of dynamic

parameters in the during the forward dynamics calculation. Fig (4.2) and (4 3) show the tracking error plots for the trajectories 1 and 2 respectively, with EKF as the state estimator, while fig (4.4) and fig (4.5) are the respective plots for the control scheme with CGEKF as the state estimator Table (4.2) shows the list of all the parameters used in these simulation studies By comparing these results with those obtained in Chapter 3 it can be observed that with the identifier in the closed loop the performance of the system has improved significantly. The maximum tracking errors and the variances of the tracking errors for the two trajectories with EKF and with CGEKF are shown in table (4 3) and (4.4) respectively. It can be observed form these results that tracking errors with CGEKF are slightly higher than the tracking errors for the control scheme with EKF From these results it can be inferred that with a slight degradation of performance, CGEKF can be used for the state estimation which reduces the computation time drastically.

Table 4.1

$P(0) = 100\, I$, $\lambda = 0.975$

where I is a 6X 6 identity matrix

Initial Values of Dynamic Parameters

$m_1 = 14$

$lc_1 = 0.46$

$I_1 = 3.75$

$m_2 = 8.6$

$lc_2 = 0.175$

$I_2 = 0.533$

variance of noise in position measurements = 0.001

variance of noise in velocity measurements = 0.05

Table 4.2

Design Parameters of the Identifier: (Refer Table 4.1)

Design parameters of the tracking controller .

Q = diag (1000, 1000, 1000, 1000)

R = diag (1, 1)

Design parameters of the EKF :

R = diag(0.001, 0 001, 0.05, 0.05)

Q = diag(0.0025, 0.0025, 0 050, 0 050)

P(0)= R

β = diag(0.9, 1.0, 1.1, 1.3)

Design parameters of the CGEKF :

R = diag(0.001, 0 001, 0.05, 0.05)

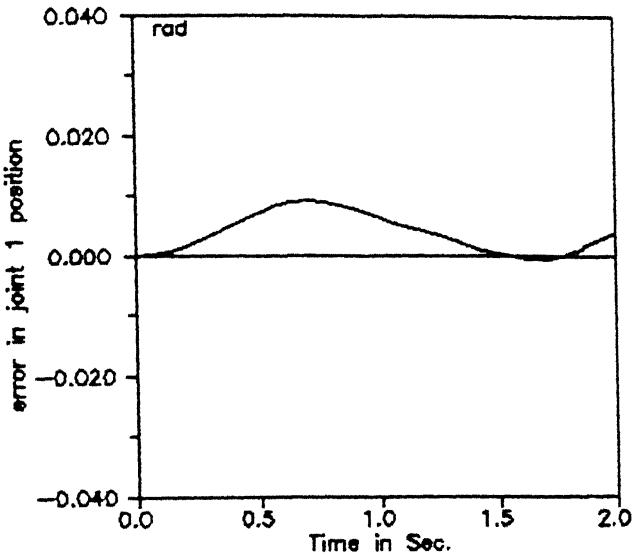
L = diag (0.01475, 0.01475, 0.225, 0 225)

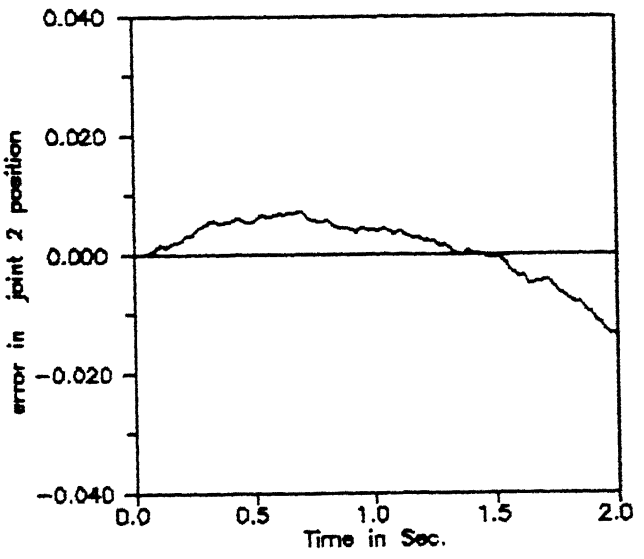
Table 4.3

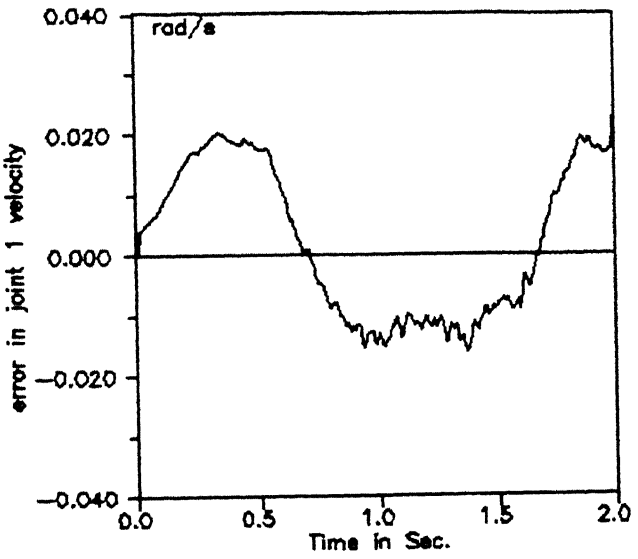
Trajectory	joint	Maximum Tracking Error with EKF		Maximum Tracking Error with CGEKF	
		position (rad)	velocity (rad/s)	position (rad)	velocity (rad/s)
Semi-Circular	1	0.00918	0.02729	0 06248	0.05252
	2	0.01394	0.07957	0.02007	0.17122
Elliptical	1	0 01429	0.05128	0.01416	0.05291
	2	0.01070	0.07410	0.01640	0.19528

Table 4.4

Trajectory	joint	Variance of Tracking Error with EKF		Variance of Tracking Error with CGEKF	
		position (rad)	velocity (rad/s)	position (rad)	velocity (rad/s)
Semi-Circular	1	0.00029	0.00110	0.00075	0.00248
	2	0.00028	0.00289	0.00025	0.00473
Elliptical	1	0.00028	0.00106	0.00019	0.00129
	2	0.00052	0.00580	0.00491	0.00602







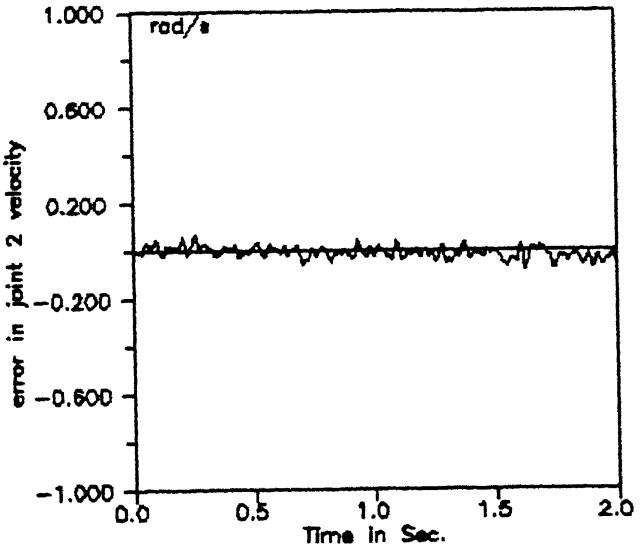
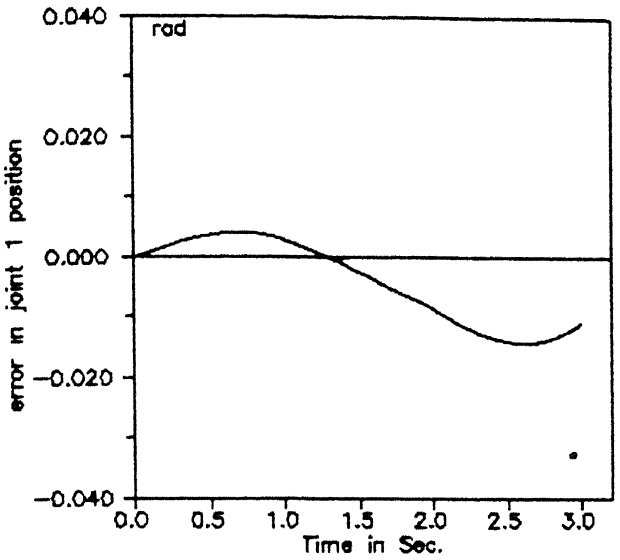
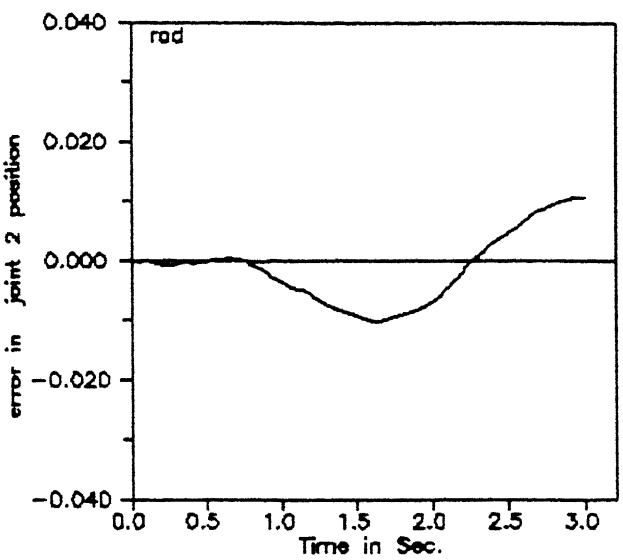
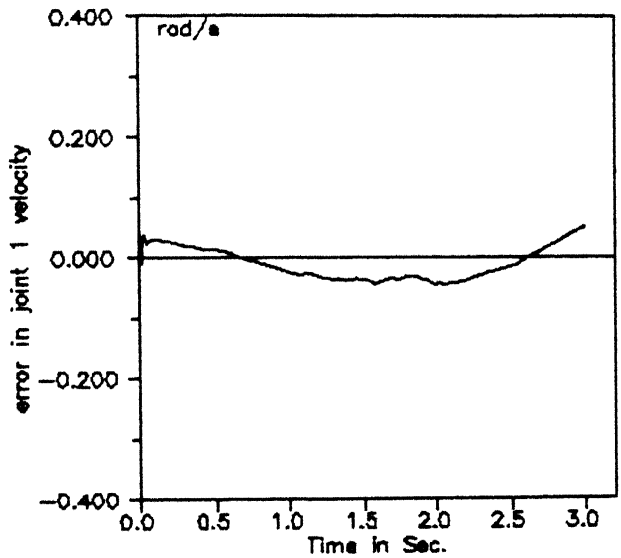


Fig. 4.2 Tracking Error Plot with EKF for Trajectory 1







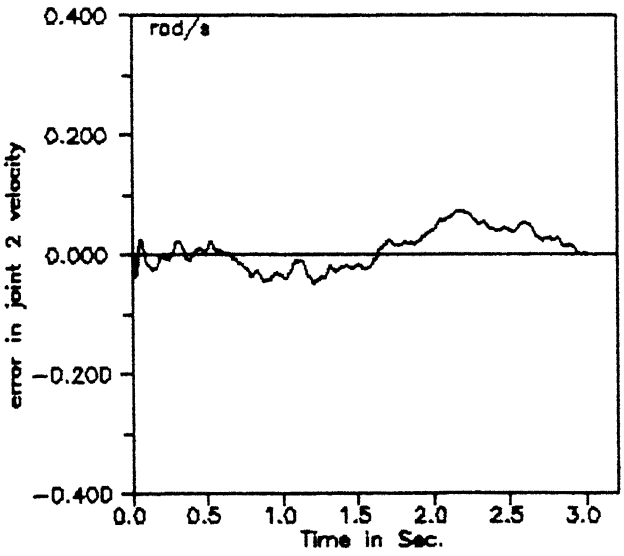
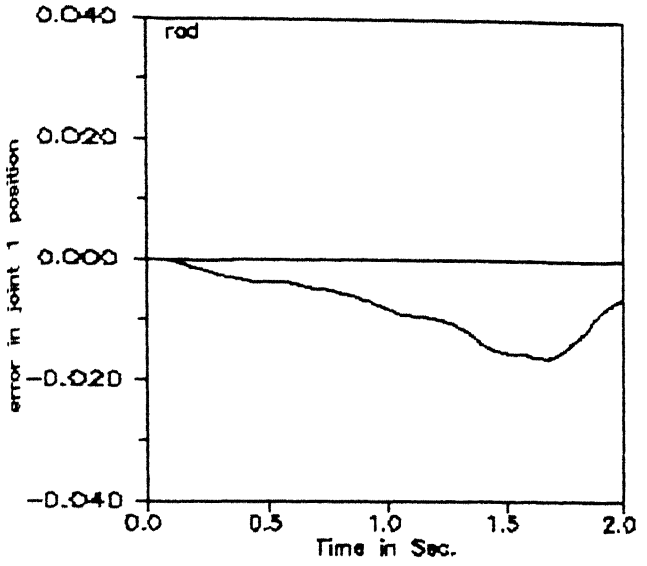
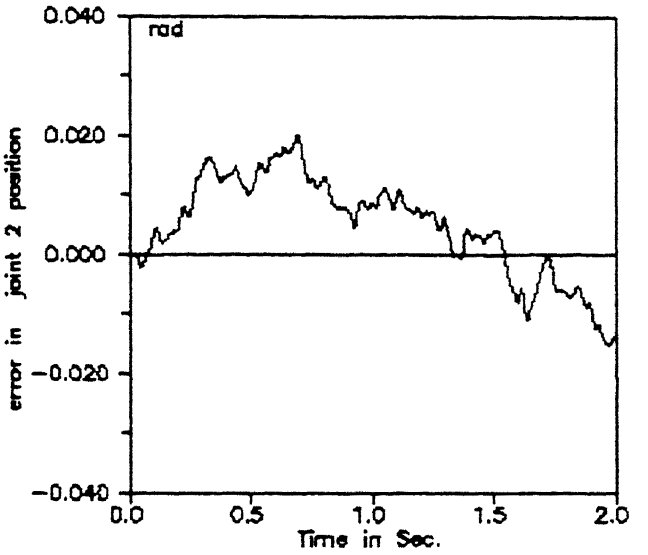
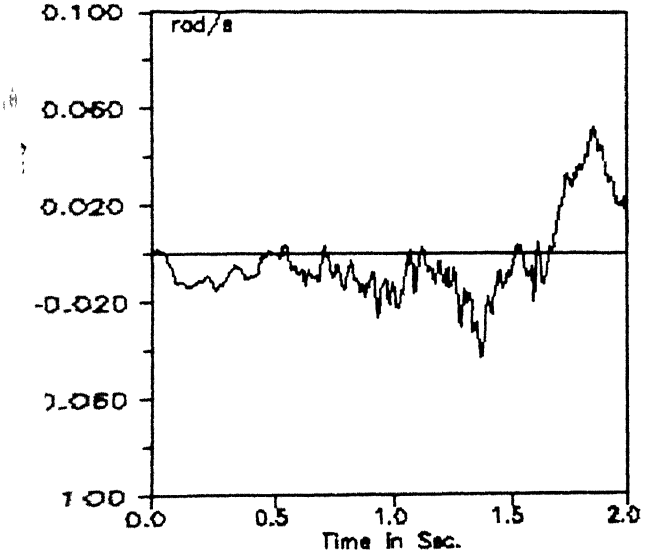


Fig. 4.3 Tracking Error Plot with EKF for Trajectory 2







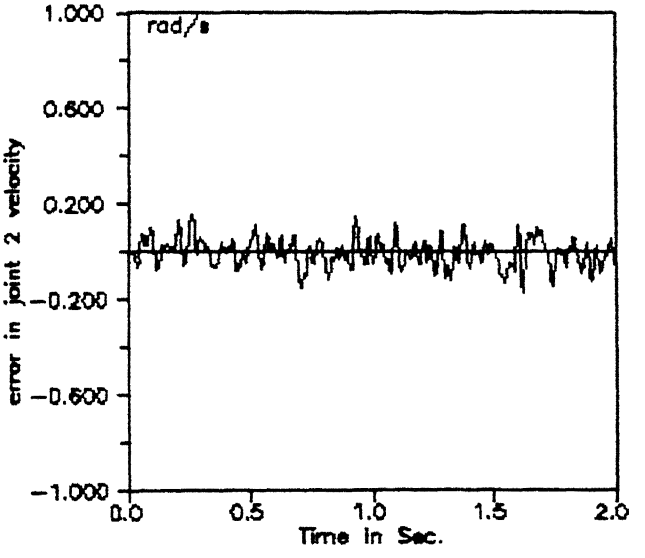


Fig. 4.4 Tracking Error Plot with CGEKF for Trajectory 1

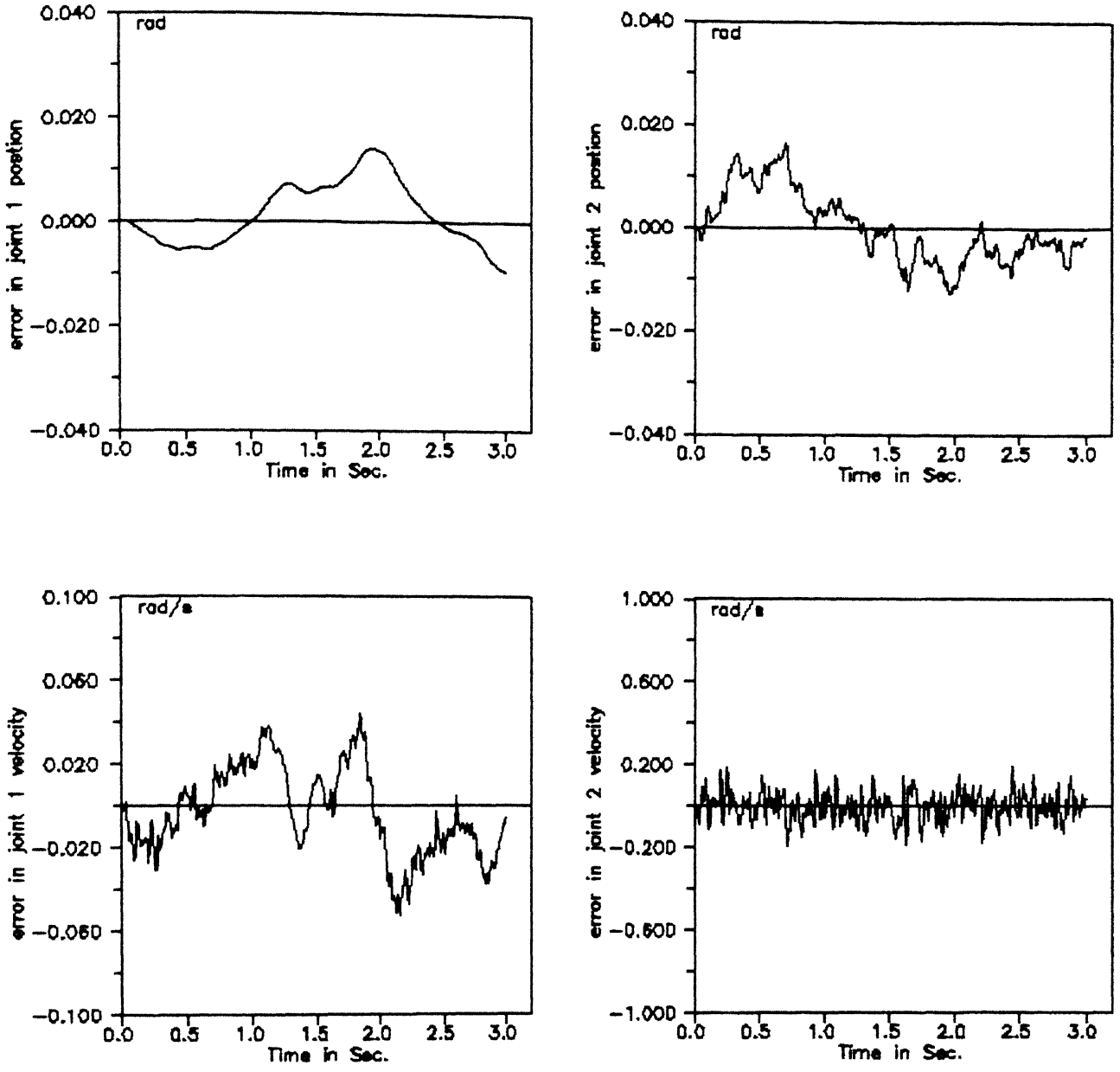


Fig. 4.5 Tracking Error Plot with CGEKF for Trajectory 2

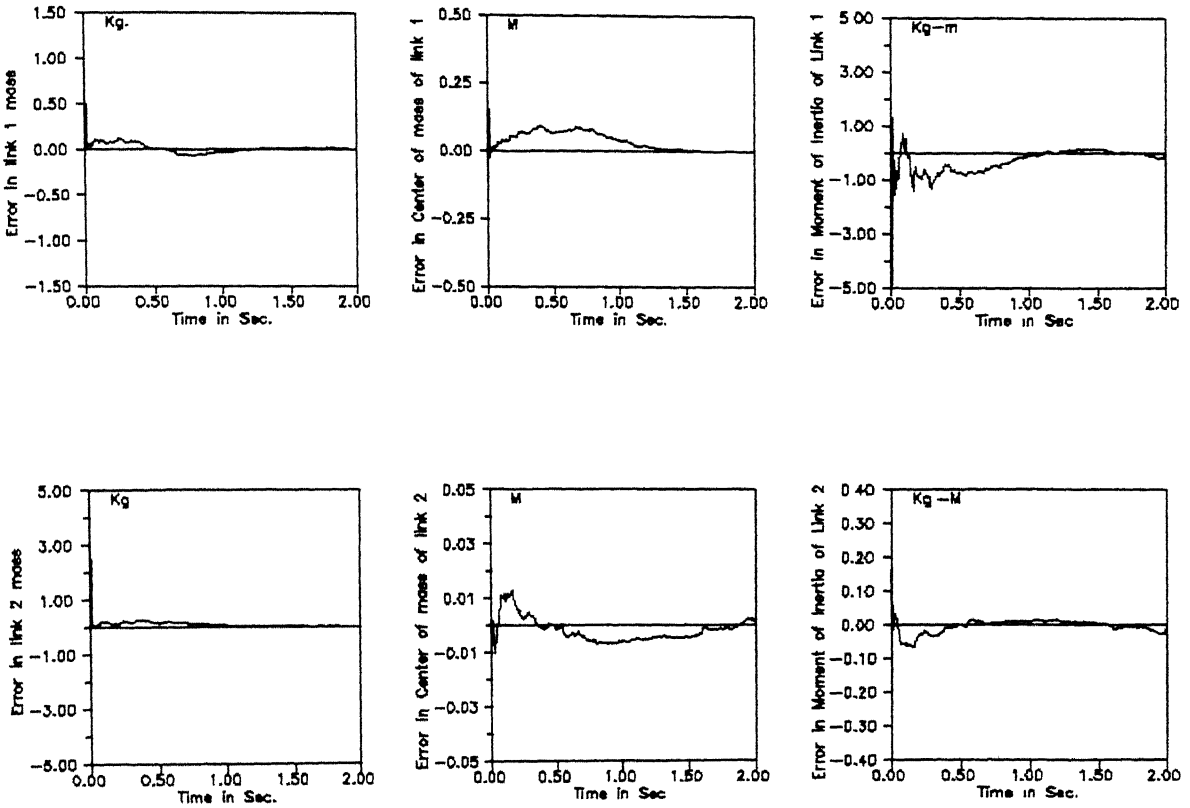


Fig. 4.2 Dynamic Parameter Estimation Errors

CHAPTER 5

Robust Computed Torque Control Schemes

In this Chapter two control schemes are discussed which incorporate a special form of feedforward term for full dynamics compensation. These control schemes differ from the control schemes discussed in the earlier chapters in the sense that they use the full nonlinear feedback compensation unlike the control schemes discussed previously which use the linearized perturbation model for feedback compensation. One of these schemes is essentially a modification of scheme [11] and the other is proposed by Majee [15]. The control schemes [15] and [11] are briefly dealt. The performance of these control schemes in presence of measurement noise is studied by conducting simulation tests on the two degree of freedom planar manipulator.

5.1. Controller With Sliding Mode :

Consider the manipulator dynamic equation (2.1). It is to be noted that in the dynamic equation the two matrices D and H are not independent of each other [11]. Given a proper representation of H, the matrix ($\dot{D}$ - 2H) is skew-symmetric. Using this fact the following equation can be derived.

$\dot{D} = H + H^T$ since is $\dot{D}$ symmetric . (5.1)

Taking into account the influence caused by unknown variation of the system parameters, the dynamic equation of the robot manipulator can be written as,

$(D+\Delta D)\,\ddot{q} + (H + \Delta H)\,\dot{q} + (g+\Delta g) = u$ (5.2)

where modeling errors are represented by ΔD, ΔH and Δg .

Depending on these parameters the following control law is suggested [28].

$$u = D(q)\,[\,\ddot{q}_r + (\Lambda\dot{q}_r + \Gamma\,\dot{q}_r + d)\,] + H(q,\dot{q}_r)\,\dot{q}_r + g(q) - K_D\,s \qquad (5.3)$$

where the vector $\dot{q}_r$ and $\ddot{q}_r$ are called reference velocity and reference acceleration respectively, and are defined as,

$$\ddot{q}_r = \ddot{q}_d - C\dot{e} \qquad (5.4)$$

$$\dot{q}_r = \dot{q}_d - Ce \qquad (5.5)$$

for a constant positive definite matrix C. The position and velocity tracking errors are defined as,

$$e = q - q_d \qquad (5.6)$$

$$\dot{e} = \dot{q} - \dot{q}_d$$

The vector s is a combination of position and velocity tracking errors and can be taken as a measure of tracking accuracy. This is defined as

$$s = \dot{e} + C\,e \qquad (5.7)$$

$$= \dot{q} - \dot{q}_r$$

Defining s as the sliding surface, to let the tracking error to zero the condition $s\dot{s}^T < 0$ should be satisfied. This ensures that the system states continue on the surface s=0. Under the assumption that sliding mode starts and continues on the surface s=0, $\forall\ q$ and $\forall\ \dot{q}$, the value of $\dot{s}$ can also be approximated to zero for all $\ddot{q}$ and $\ddot{q}_d$. Thus, from (5.7),

$$\dot{q} = \dot{q}_r \qquad (5.8)$$

$$\ddot{q} = \ddot{q}_r \qquad (5.9)$$

Eqns. (5.5) and (5.9) are combined to obtain

$$D\,(\ddot{e} + C\,\dot{e}) = D\,[\,(\Lambda - \Delta D')\,\ddot{q}_r + (\Gamma - \Delta H')\,\dot{q} + (d - \Delta g')\,] = u \qquad (5.10)$$

Substituting the eq. (5. 7) in eq. (5.11)

$$\dot{s} = (\Lambda - \Delta D')\,\ddot{q}_r + (\Gamma - \Delta H')\,\dot{q}_r + (d - \Delta g') \qquad (5.11)$$

where $\Delta D' = D^{-1}\Delta D$, $\Delta H' = D^{-1}\Delta H$ and $\Delta g' = D^{-1}\Delta D$

Now, for sliding mode to occur $s_i\dot{s}_i < 0$ for i = 1 , 2 .....n. By using eq. (5.11) the following relationship can be obtained.

$$\sum_{j=1}^{n} (A_{ij} - \Delta D') \ddot{q}_{rj} s_i + \sum_{j=1}^{n} (H_{ij} - \Delta H') q_{rj} s_i + (d_i - \Delta g_i) s_i < 0 \qquad (5.12)$$

The goal is to find A_{ij} and Γ_{ij} and d_i so that the above inequality is always valid. They are chosen as follows,

$$A\lambda_{ij} = -\Delta 1_{ij} \text{ sign } (s_i \ddot{q}_{rj}) \qquad (5.13)$$

$$\Gamma_{ij} = -\Delta 2_{ij} \text{ sign } (s_i \dot{q}_{rj}) \qquad (5.14)$$

$$d_i = - \eta 1_i \text{ sign } (s_i) \qquad (5.15)$$

where $\Delta 1$ and $\Delta 2$ are positive definite matrices and $\eta 1$ is a vector with all positive entries as defined below.

$\Delta 1$ = maximum variation in D ' matrix and is known.

$\Delta 2$ = maximum variation in H' matrix and is known .

$\eta 1$ = maximum variation in g' vector and is known.

For checking the stability of the control scheme, consider the following Lyapunov function.

$$V(t) = 1/2 \, s^T D \, s \qquad (5.16)$$

Differentiating V(t) with respect to time ,

$$\dot{V}(t) = 1/2 \, s^T \dot{D} \, s + s^T D \, \dot{s}$$

$$= 1/2 \, s^T \dot{D} \, s + s^T D [\ddot{q} - \ddot{q}_r]$$

$$= s^T/2 [(\dot{D} - 2H)s + 2 (u - H\dot{q}_r - g - D\ddot{q}_r)]$$

Cconsidering the conservation of energy in the manipulator, the following relationship can be derived [11],

$$1/2 \, d/dt [\dot{q}^T D \, q] = \dot{q} (u - g)$$

where the left hand side is the derivative of the manipulator's kinetic energy and the right hand side represents the power input from the actuators .

Differntiating the right-hand side explictly

$$\dot{q} (u - g) = \dot{q}^T D \ddot{q} + 1/2 \, \dot{q}^T \dot{D} \dot{q}$$

and expanding the term $D\ddot{q}$ by using eq. (2.1) ,

$$\dot{q}^T (\dot{D} - 2 H) \dot{q} = 0$$

and from this it is clearly observed that

$$\dot{s}^T (\dot{D} - 2H) \dot{s} = 0$$

Hence,

$$\dot{V}(t) = s^T [u - H \dot{q}_r - g - D\ddot{q}_r] \qquad (5.17)$$

Now replacing u with the control law given in eq. (4) , then eq. (14) becomes,

$$\dot{V}(t) = s^T D^{-1} [(A - \Delta D') \ddot{q}_r + (\Gamma - \Delta H') \dot{q}_r + (d - \Delta g)] - s^T K_D s$$

From eqns.(5.13), (5.14) and (5.15) it is evident that the first term within the square brackets is always negative for any s except s=0 . and also since K_D is positive definite, $\dot{V}(t)$ is always negative definite. Hence the above formulation leads to the global asymptotic convergence of the tracking error Now due to the presence of sign(s) term in the control signal there will be excessive control activity even in the case of small error. To cope with this problem a function f(s) instead of sign(s) defined as below, is used.

$$f(s_i) = \begin{cases} 1 & \text{if } s_i > \phi \\ s_i/\phi & \text{if } \phi < s_i < \phi \\ -1 & \text{if } s_i \leq \phi \end{cases} \qquad (5.18)$$

ϕ is called the control bandwidth and this is totally dependent on designer's choice. So finally, the control law becomes,

$$u = D(q) [\ddot{q}_r + (A\dot{q}_r + \Gamma \dot{q}_r + d)] + H (q , \dot{q}_r) \dot{q}_r + g (q) - K_D s \qquad (5.19)$$

with

$$A_{ij} = -\Delta 1_{ij} f(s_i \ddot{q}_{rj}) \qquad (5.20)$$

$$\Gamma_{ij} = -\Delta Z_{ij} f(s_i \dot{q}_{rj}) \qquad (5.21)$$

$$d_i = - \eta 1_i f(s_i) \qquad (5.22)$$

Using this formulation Majee [15] designed a control scheme in which the noisy feedback measurements are filtered using the Extended Kalman Filter theory.

From the simulation studies of the above control scheme Majee remarked that under high amount of modeling imperfection, the sliding mode control gives rise to control chattering which may excite the high frequency unmodelled

dynamics. In order to overcome this difficulty a control scheme in which the robot dynamic parameters are identified as a part of the control objective i.e., to force the error between the actual and desired state trajectories to zero is studied below. Fig (5.1) shows the block diagram of the overall control scheme in corporating the sliding mode control and the Extended Kalman Filter to disambiguate the noisy feedback measurements

5.2 Adaptive Control Scheme :

This scheme exploits the fact that the robot dynamic equation can be recasted such a way that it is linearly related to a particular set of equivalent robot dynamic parameters. The dynamic equation of the robot model, eq (2. 1) can be rewritten as shown below.

$$D(\underline{q})\,\ddot{\underline{q}} + H(\underline{q},\dot{\underline{q}})\,\dot{\underline{q}} + g(\underline{q}) = W(\ddot{\underline{q}},\dot{\underline{q}},\underline{q})\,\underline{p} \qquad (5.23)$$

where $\underline{p}$ is m dimensional vector containing the unknown and recast dynamic parameters, and W is n$\times$ m matrix independent of the vector $\underline{p}$.

Using the above equation Slotine and Li [11] suggested the following control law,

$$u = \hat{D}(q)\,[\,\ddot{q}_r + \hat{H}(q,\dot{q}_r)\,\dot{q}_r + \hat{g}(q)\,] - K_D\,s \qquad (5.24)$$

$$\dot{\hat{p}} = -\Gamma\,W^T s \qquad (5.25)$$

where Γ is a constant positive definite matrix

To prove the global asymptotic stability of the above control law the following Lyapunov function is considered.

$$V(t) = 1/2\,[s^T D\,s + \tilde{p}^T\,\Gamma^{-1}\,\tilde{p}\,] \qquad (5.26)$$

Differntiating V(t) with respect to time and proceeding in similar lines to that of the derivation for the sliding mode controller the following equation can be obtained.

$$\dot{V}(t) = s^T(\,\hat{D}\,\ddot{q}_r + \hat{H}\,\dot{q}_r - g - K_D s\,) + \tilde{p}^T\,\Gamma^{-1}\,\dot{\tilde{p}}$$

$$= s^T[\,W\,\tilde{p} - K_D s\,] + \tilde{p}^T\Gamma^{-1}\,\dot{\tilde{p}}$$

Substituting eq. (5 25) in the above it can be shown that

$$\dot{V}(t) = -s^T K_D s \leq 0 \qquad (5.27)$$

which shows the global asymptotic convergence of the controller.

In order to take into account the noisy feedback measurements the EKF algorithm discussed in the previous chapters is used. The overall block diagram of the control scheme is shown in fig (5.2).

5.3 Simulation Experiment :

The performance of these control scheme are studied with the help of the simulated two degree of freedom planar manipulator The manipulator dynamic equation (2.1) is parameterized as given below.

$$u_1 = I_1' \ddot{q}_1 + I_2' (\ddot{q}_1 + \ddot{q}_2) + m_2 [l_1^2 \ddot{q}_1 + g l_1 \cos (qi)]$$

$$+ lc_2' [2l_1 \cos (q_2) \ddot{q}_1 + l_1 \ddot{q}_2 \cos(q_2) - l_1 \sin (q_2) \dot{q}_2^2 - 2 l_1 \sin (q_2) \dot{q}_1\dot{q}_2 + g\cos(q_1+q_2)]$$

$$+ lc_1' [2 \ddot{q}_1 \cos(q_1) + l_1 \sin (q_2) \dot{q}_1\dot{q}_2 + g \cos(q_1)]$$

$$u_2 = I_2' (\ddot{q}_1 + \ddot{q}_2) + lc_2' [l_1 \ddot{q}_1 \cos (q_2) + l_1 \sin(q_2) \dot{q}_2 \dot{q}_2 + g \cos (q_1 + q_2)$$

where

$$I_1' = I_1 + m_1 lc_1^2$$

$$I_2' = I_2 + m_2 lc_2^2$$

$$lc_1' = m_1 lc_1$$

$lc_2' = m_2 lc_2$ are the equivalent dynamic parameters. With this formulation the equivalent parameter vector p is given by

$p^T = [I_1' \;\; I_2' \;\; m_2 \;\; lc_2' \;\; lc_1' \;\;]^T$ and the elements W_{ij} i=1..5, j=1..2 of the W matrix are given by

$$W_{11} = \ddot{q}_1$$

$$W_{12} = (\ddot{q}_1 + \ddot{q}_2)$$

$$W_{13} = l_1^2 \ddot{q}_1 + g l_1 \cos (q_1)$$

$$W_{14} = 2 l_1 \cos (q_2) \ddot{q}_1 + l_1 \ddot{q}_2 \cos(q_2) - l_1 \sin (q_2) \dot{q}_2^2 - 2 l_1 \sin (q_2) \dot{q}_1\dot{q}_2 + g \cos(q_1+q_2)$$

$$W_{15} = g \cos (q_1)$$

$$W_{22} = \ddot{q}_1 + \ddot{q}_2$$

$w_{24} = l_1 \ddot{q}_1 \cos(q_2) + l_1 \sin(q_2) \dot{q}_2^2 + g \cos(q_1 + q_2)$

The rest of the elements of W are zero.

The matrix Γ is chosen to be an Identity matrix.

Regarding the state estimator, all the arguments made in the previous chapters are valid for this case also. The nominal trajectory in this case is assumed to be a circular path having constant acceleration, uniform velocity and constant deceleration regions respectively The trajectory is generated using the procedure described in chapter 2. The duration is of this trajectory is assumed to be 4 sec. The nominal joint trajectories are shown in fig. (5.3) The simulation experiment is performed at different load conditions with a maximum load of 4 Kg. attached to the second link. The performance this control scheme is compared with that of the sliding mode control scheme proposed in [15]. In table (5.1) the values of the parameters used in this simulation studies are shown. Fig (5.4) and fig. (5.5) show the tracking error plots for full load with identifier and with sliding model control respectively. Similarly, fig. (5.6), (5.7), (5.8) and (5.9) show the respective plots for half load and noload conditions. Table (5.2) and (5.3) show the maximum tracking errors and the variance of the tracking errors for the two control schemes discussed in this chapter for the three different loading conditions. It can be observed from these results, both the schemes result in almost the same performance and the performance of the adaptive control scheme is marginally better than that of the nonadaptive scheme in the case of large parametric variations.

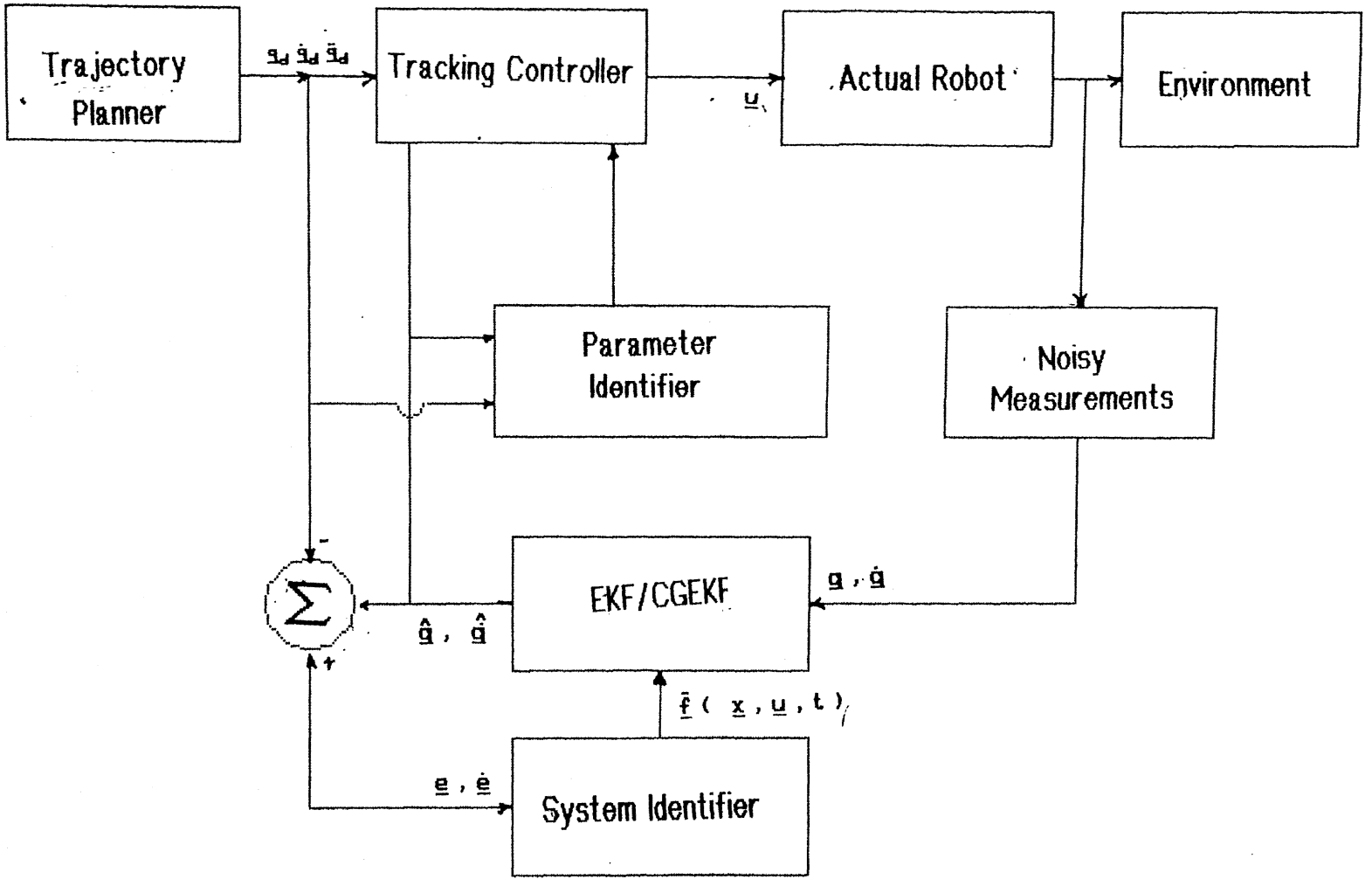


Fig. 5.2. Block Diagram of the Adaptive Control Scheme

Table 5.1

Design Parameters of the tracking controller :

K_D = diag (400, 300)

C = diag (8, 8)

For sliding Mode control :

ϕ = 0.01 unit

$\Delta 1$, $\Delta 2$ and $\eta 1$: calculated from the nominal dynamic Parameters given in Table (3.1)

Parameters of the Filter :

R = diag (0.001 , 0.001, 0.05, 0.05)

Q = diag (0.003, 0.003, 0.005, 0.005)

P(0) = R

β = diag (0.9, 0.9, 1.0, 1.3)

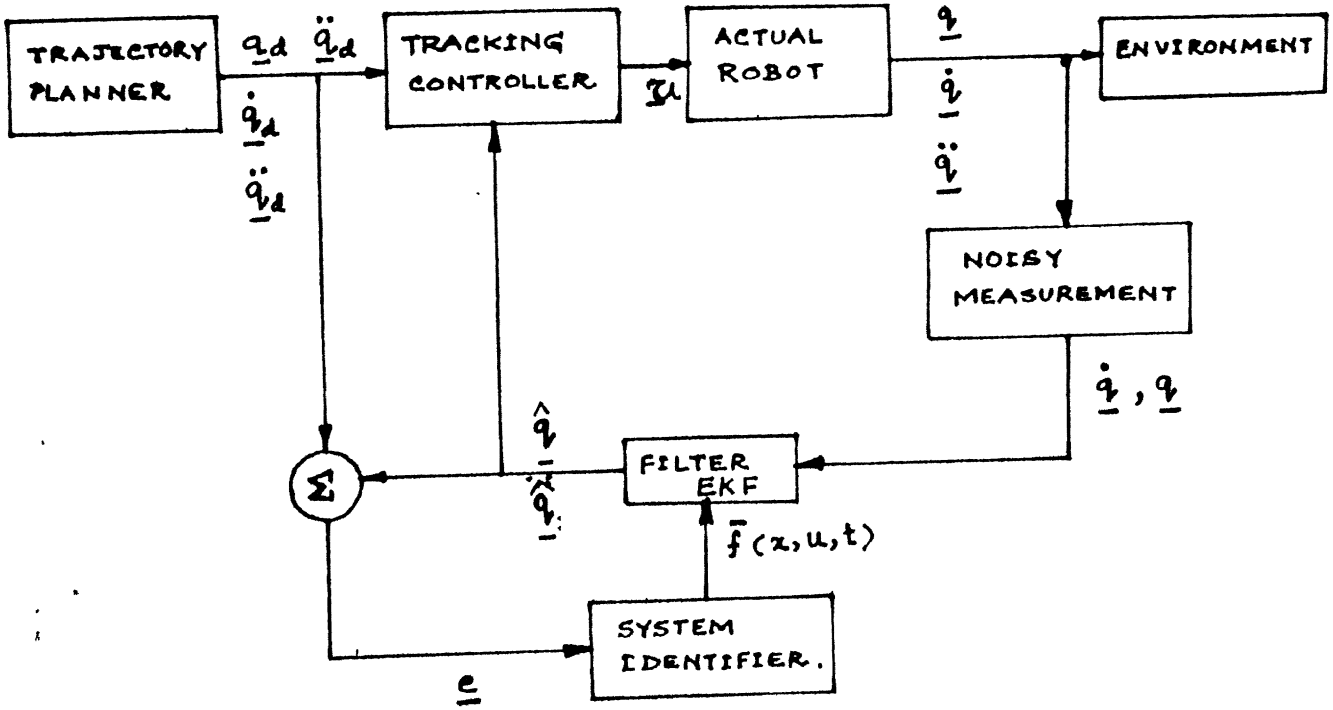
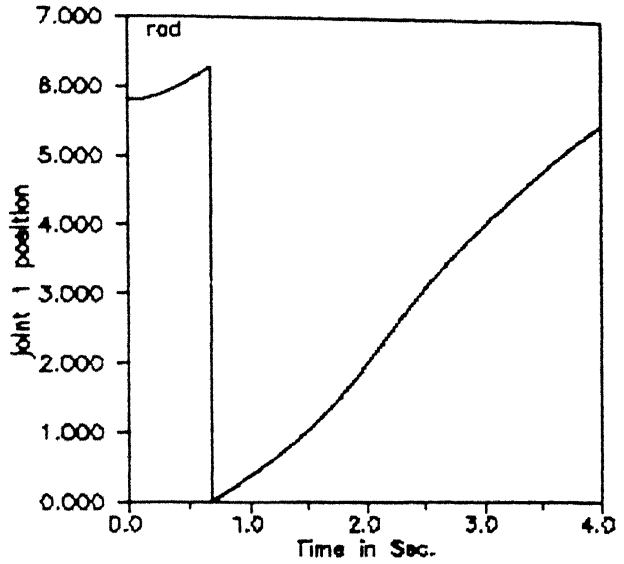
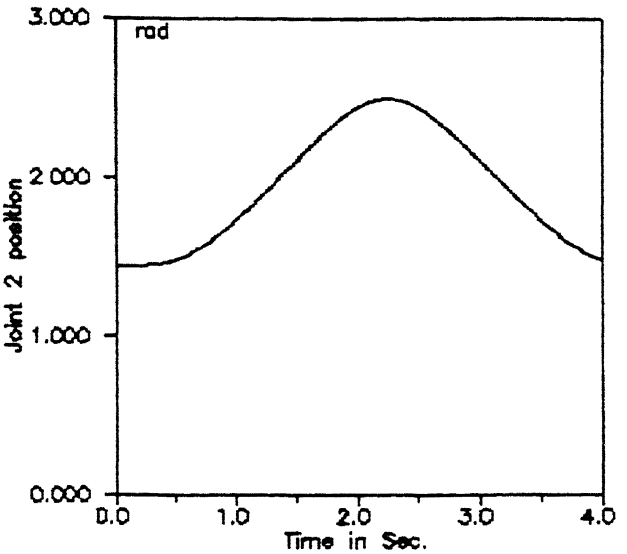
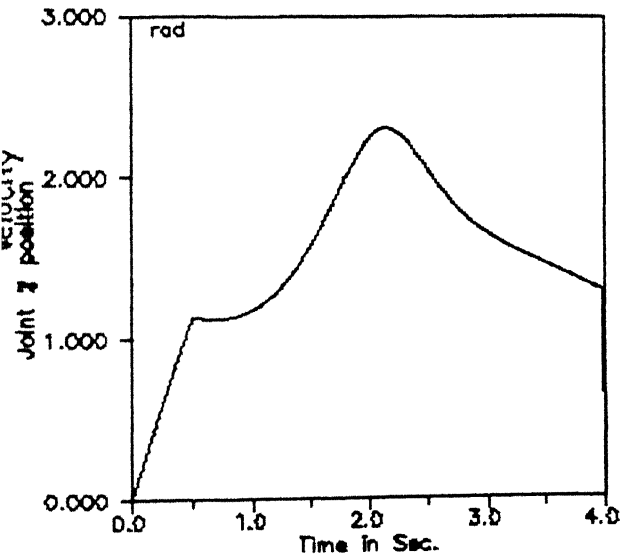


FIG 5.1 COMPLETE BLOCK DIAGRAM OF CONTROL METHOD USING SLIDING MODE







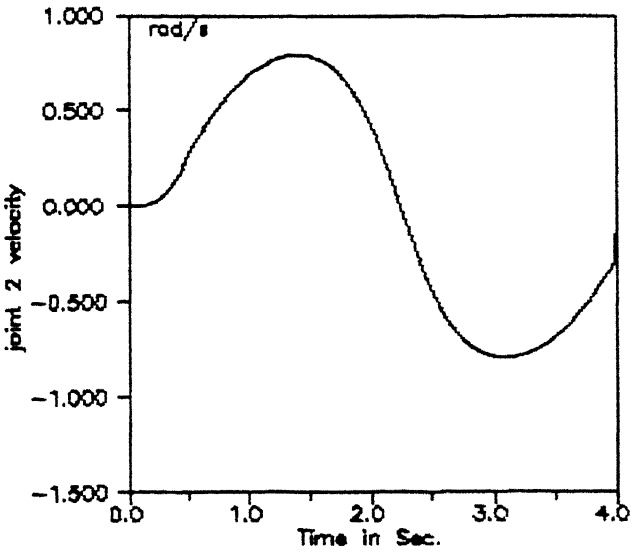
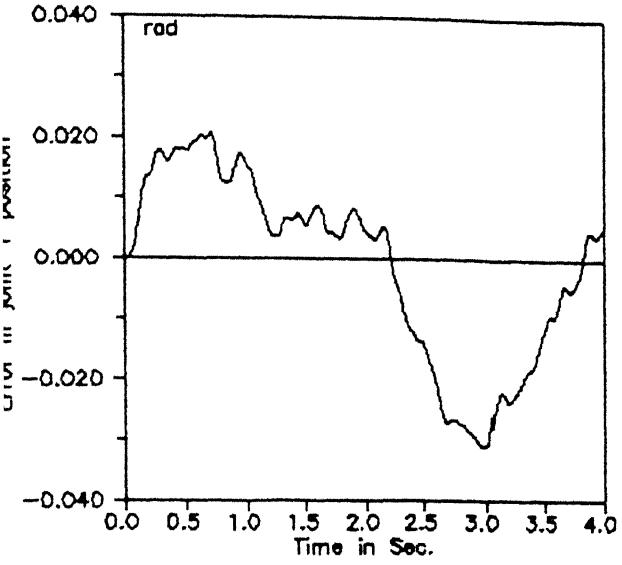
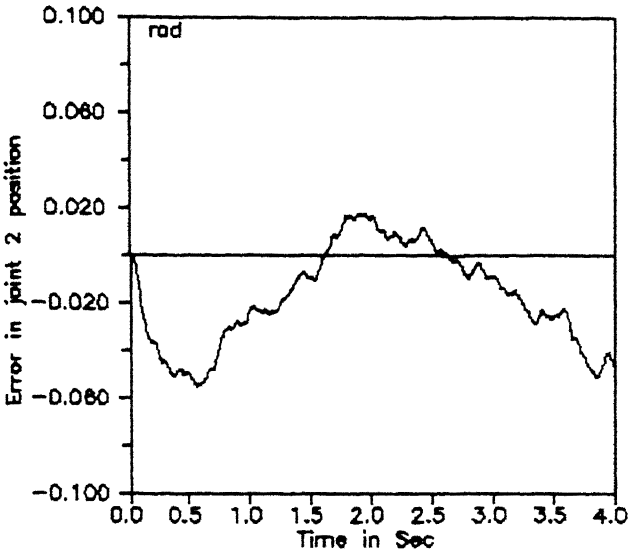
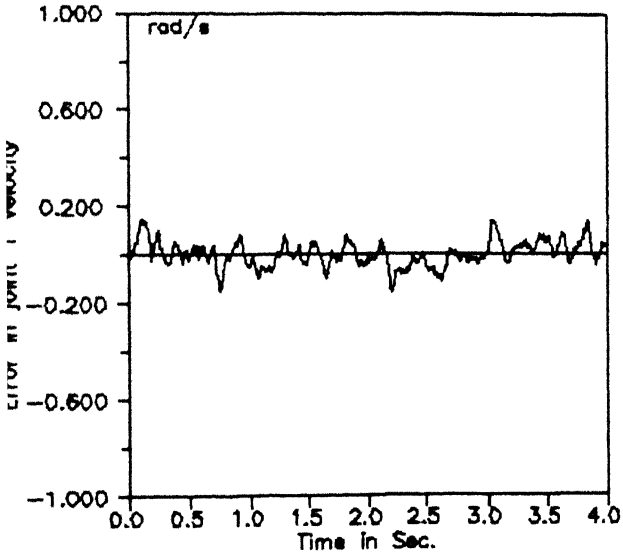


Fig. 5.3 Nominal Trajectory







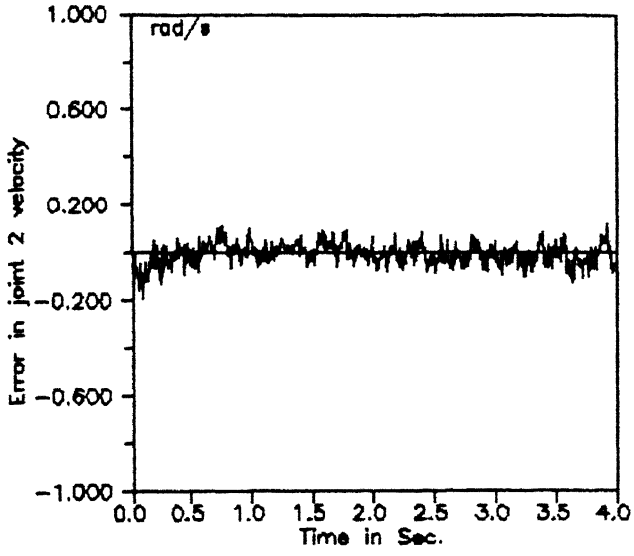


Fig. 5.4 Tracking Error With Identifier for Full Load

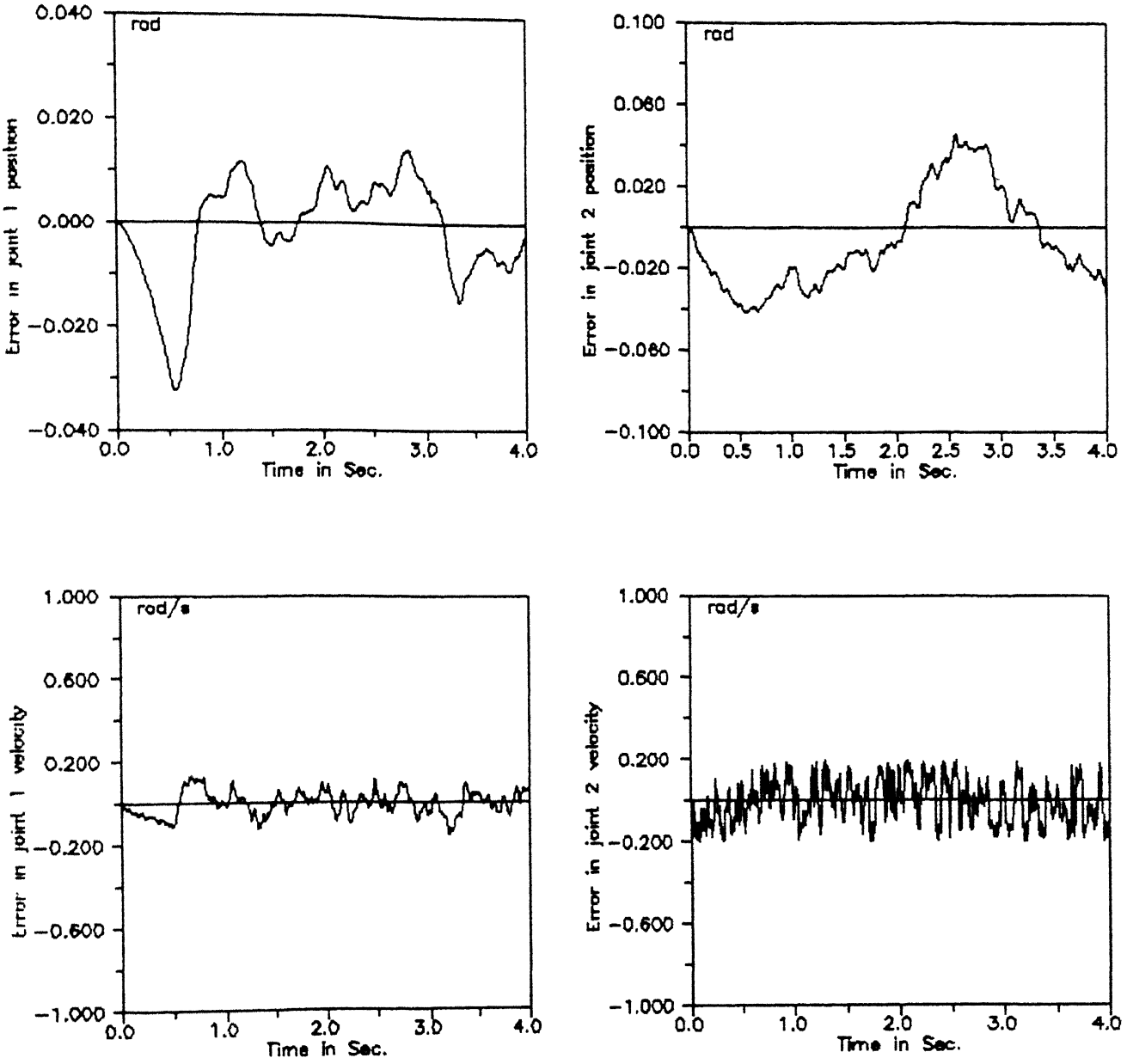


Fig. 5.5 Tracking Error With Sliding Mode for Full load

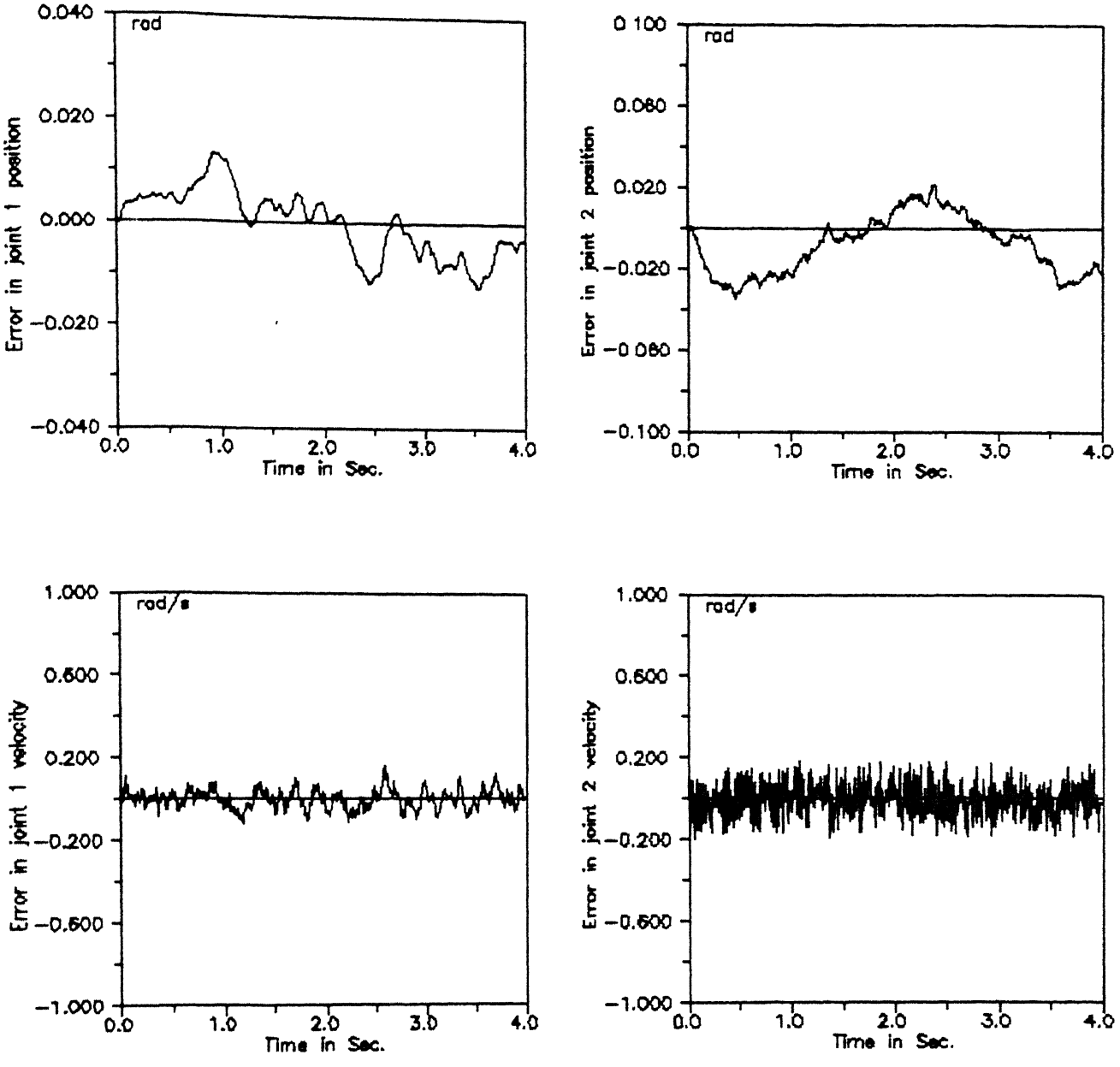


Fig. 5.6 Tracking Error With Identifier for Half load

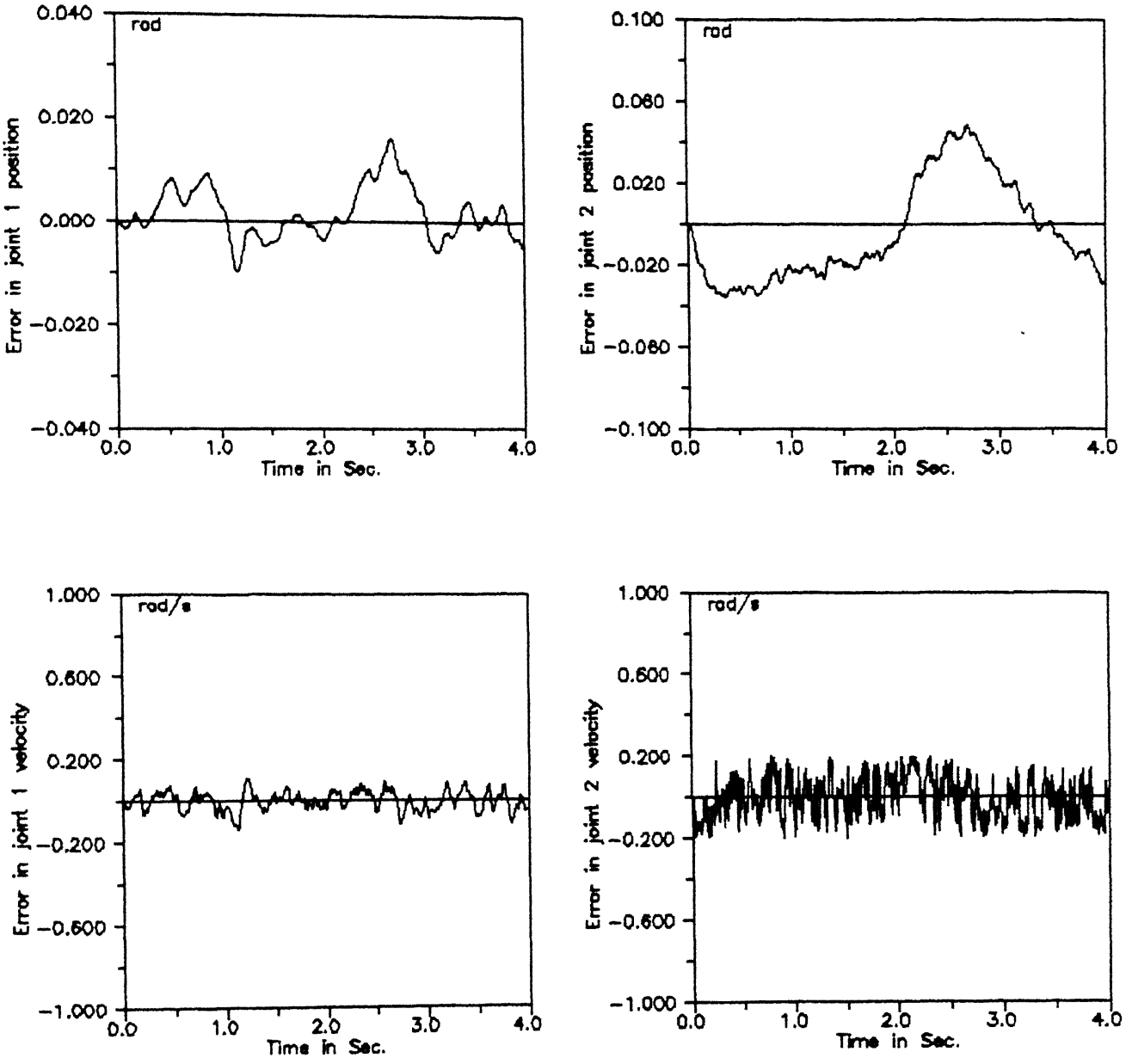


Fig. 5.7 Tracking Error With Sliding Mode for Half load

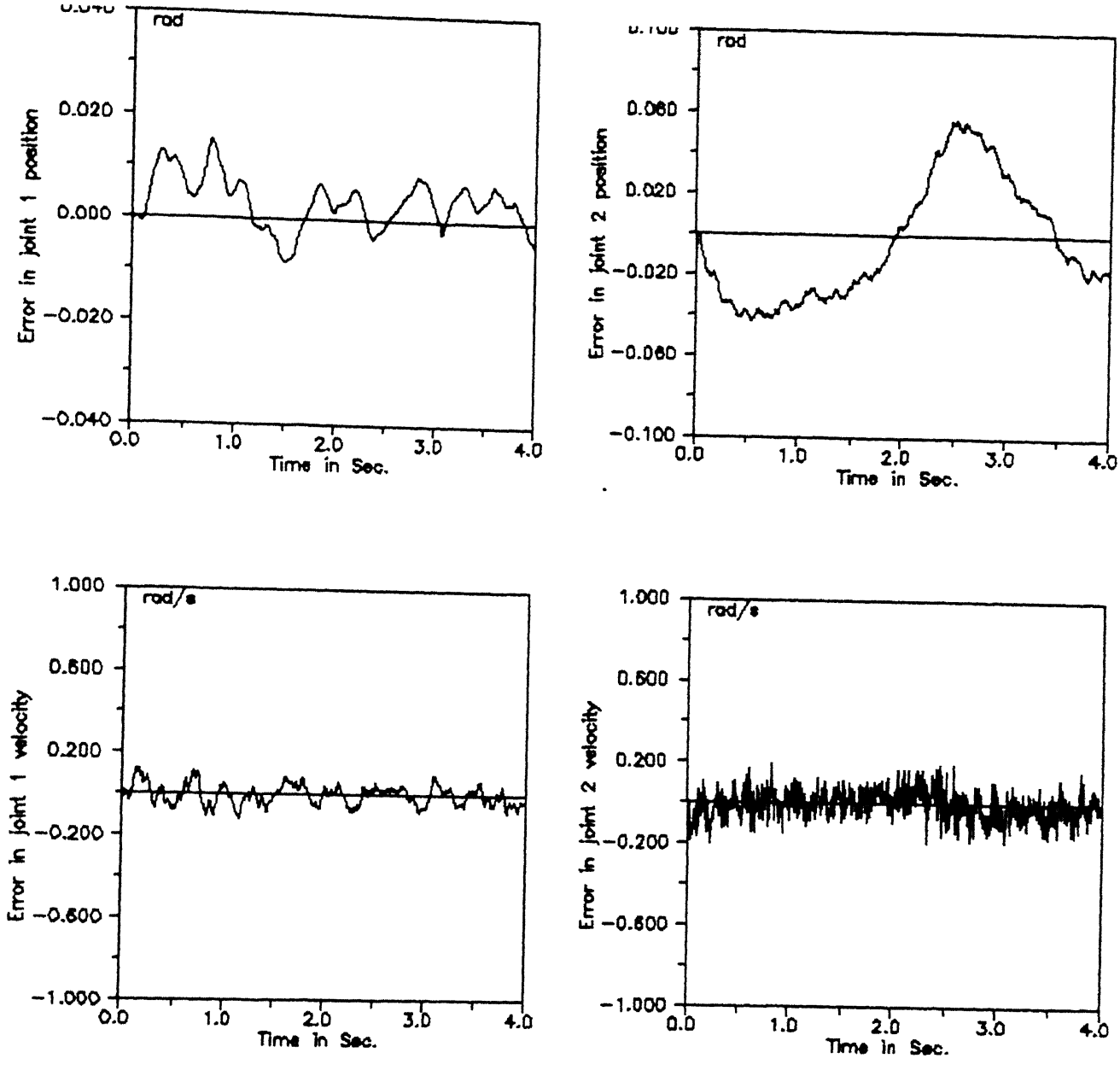
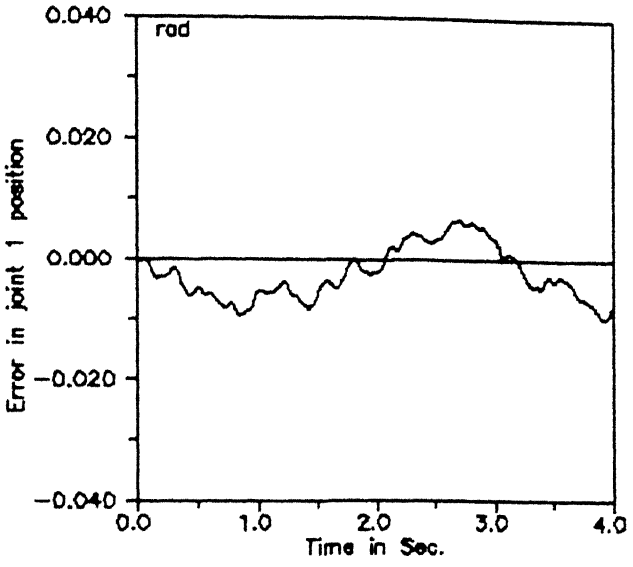
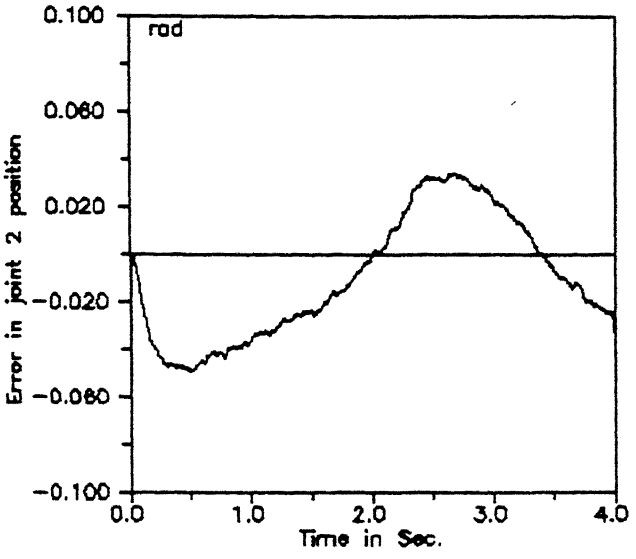
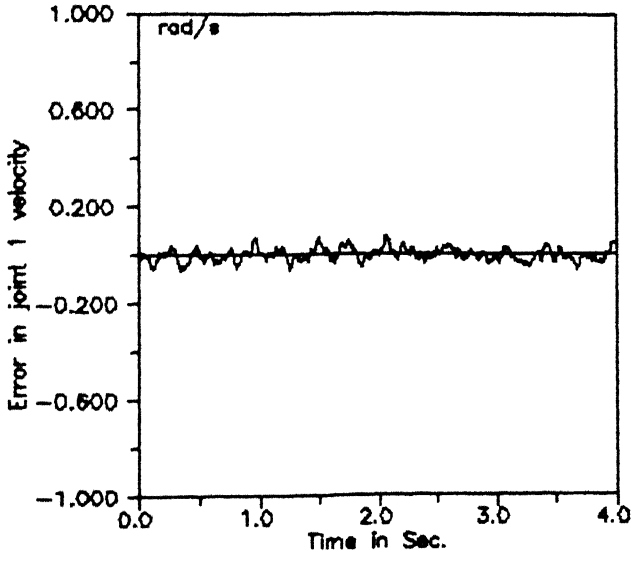


Fig. 5.8 Tracking Error With Identifier for No Load







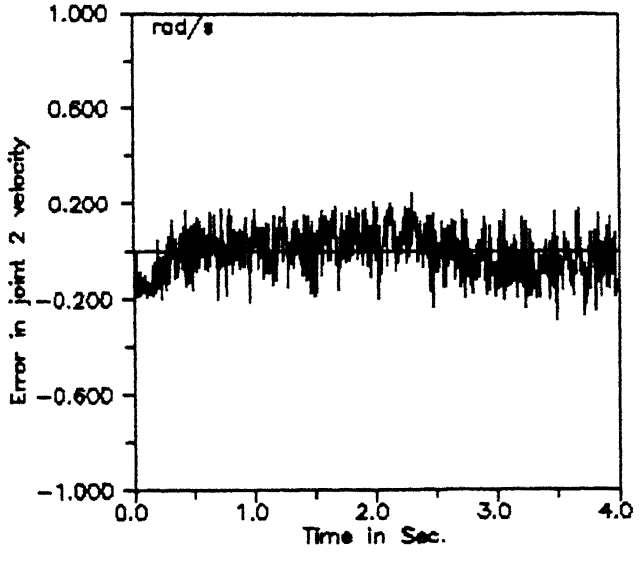


Fig. 5.9 Tracking Error With Sliding Mode for No Load

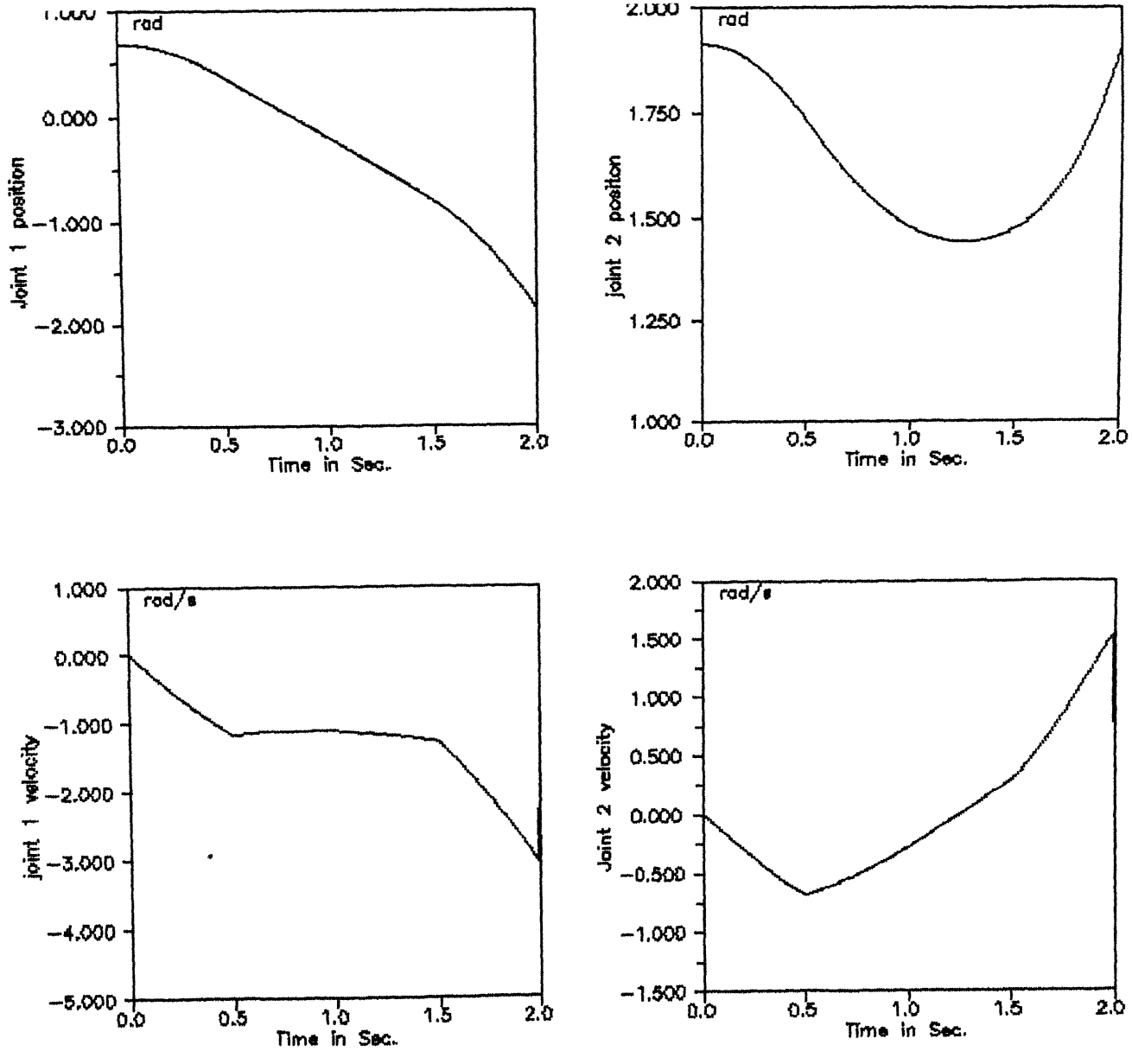


Fig. 2.4 Nominal Trajectory 1

CHAPTER 6

Conclusions

The performance of the control schemes designed in this thesis is evaluated and the practical implementation details of the actual control scheme are also presented. Finally scope for further work in this direction is suggested and the thesis concludes with the summary of ideas presented.

6.1 Comparison of results :

The results obtained in Chapter 4 and Chapter 5 can be compared. From the Table. (4.3) and (5.2) it can be observed that the tracking errors with LQ controller are smaller by one order. But these results can not be compared in absolute numerical terms, since it is expected that the simulation programs were not identical in terms of nominal trajectories. This can be due to the fact that for the nonlinear controller the control law is based on fixed PD feedback whose gains are chosen by intuition where as in the LQ case the gain matrix is calculated through minimizing the quadratic performance index.

The control scheme based on linearized perturbation equation has the disadvantage that in the formation of perturbation equations is is assumed that the tracking errors are small and hence the perturbation equations are valid only in the neighborhood of the nominal operating point. In the case of any large disturbance the control scheme may fail. the calculation of Jacobian matrices A and B , and the solving Ricatti equation on line restricts the application of this control scheme. The next section deals with this problem.

6.2 Implementational Aspects :

The various computational aspects regarding the actual implementation of the control scheme are dealt here. The control scheme essentially consists of a feedforward term, feedback term, identifier to estimate the dynamic parameters and a filter to disambiguate the feedback measurements. The overall control scheme is computationally cumbersome and it needs some modifications to physically implement the scheme which are discussed below.

It can be observed that the feedback and feedforward terms are totally independent of each other. So two separate microprocessors can be used to calculate these terms simultaneously.

In practice the parameters of the robot can be measured or estimated before hand. Since the load is usually connected to the last link, only the parameters of the load are unknown. So instead of identifying the whole set of dynamic parameters, only load parameters can be identified online reducing the computational burden.

The matrices **A, B** may be updated at a low rate while a high rate can be used to update the terms $\mathbf{g}$, $\dot{\mathbf{g}}$, $\tilde{\mathbf{q}}$. Further the identification can also be performed at low rates.

Instead of solving the Ricatti equation online one step ahead optimal control law can be used which does not require the solution of the Ricatti equation.

6.3 Scope for Further Work :

The work done in this thesis is purely simulation and no theoretical proofs are given regarding the convergence properties of the modified EKF algorithm and the the robustness properties of the control scheme and the identification scheme. So studies in this direction may give some interesting results.

The control scheme discussed in applicable only for the joint position control and many of the modern day robots are required to perform complicated tasks which require control of not only the joints positions, velocities but also the force exerted by the manipulator on the environment. The feasibility of the present control scheme for joint position and force control is to be investigated.

6.4 Conclusions:

It has been shown with the aid of illustrative examples that the LQ control scheme can be successfully utilized to stabilize and control a robot manipulator. The control of two degree of freedom planar manipulator is presented. This may be easily extended to consider multiple joint control.

Two different nominal trajectories are chosen for simulation studies to determine the capability of the controller to cope with different operating points. In both the cases the tracking errors are small demonstrating the versatility of the controller.

Models of Columb friction can also be included in the dynamic model of the robot and the corresponding coefficients can be identified.

APPENDIX A

A.1 Newton-Euler Formulation of Equations of Motion :

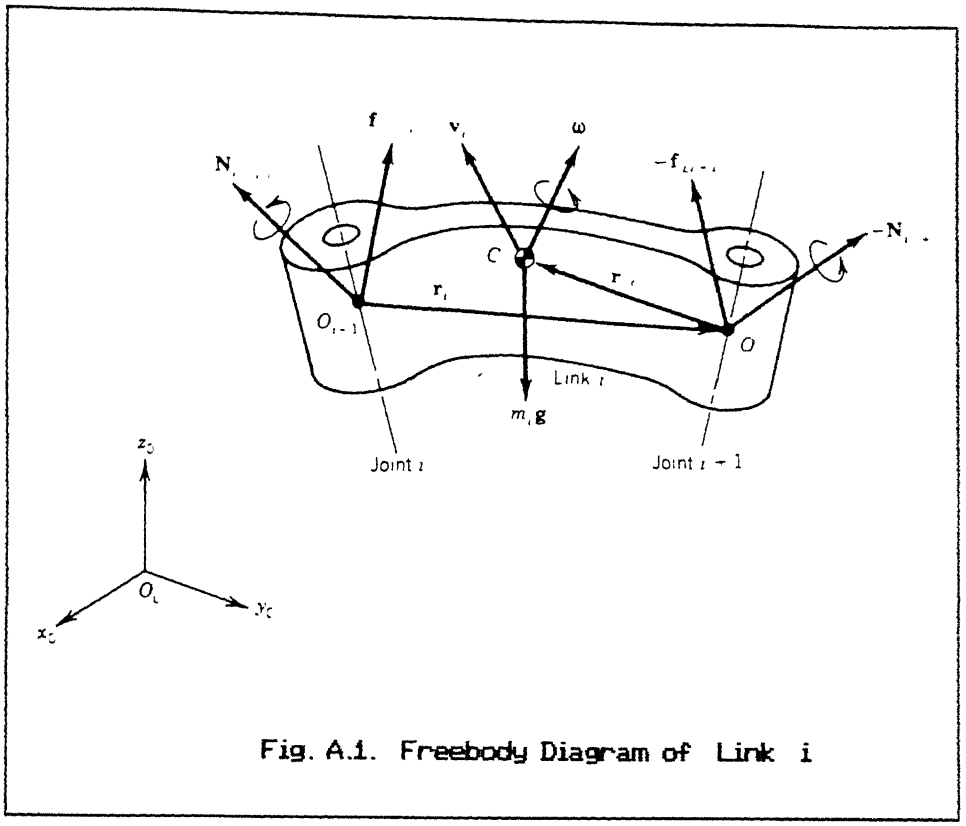


Fig. A.1. Freebody Diagram of Link i

Consider the free body diagram of a robot ,shown in fig (A1)

Let $f_{i-1,i}$ be defined as the force exerted by the $(i-1)^{th}$ link on the i^{th} link, m_i is the mass of the link i , v_{Ci} is the linear velocity of the centroid C_i of link i w.r.t a fixed coordinate system. (see fig) The balance of linear forces then yields,

$$f_{i-1,i} - f_{i,i+1} + m_i g - m_i \dot{v}_{Ci} = 0 \qquad i = 1 \ldots n \qquad (A.1)$$

Where all vectors are defined w.r.t. a base coordinate frame and there are n links.

Similarly if $N_{i-1,i}$ is defined as the movement applied to link i by link (i-1), $r_{i-1,i}$ is the position vector from point O_{i-1} to O_i in fig(A.1) , ω_i is the angular velocity vector and I_i is the centroidal moment of inertia of link i , then the balance of moments can be derived as

$$N_{i-1,i} - N_{i+1,i} - r_{i-1}\, C_i \times f_{i-1,i} + r_i\, C_i \times f_{i+1,i} - I_i \dot{w}_i - w_i \times (I_i w_i) = 0 \qquad (A.2)$$

$$i = 1, \ldots n$$

Eqns. (A.1)and (A 2) can be modified so that they are expressed in terms of input variables which can be measured . Choosing the joint torque u as the input and joint displacements $g = [\, q_1\ q_2\ \ldots\ldots\ q_n]^T$ as the outputs that locate the whole arm we derive the dynamic equations which correspond to a two degree of freedom planar manipulator [fig(2.1)] with two individual links are given below. The equations are interms of joint torques u_1 and u_2 and joint displacements q_1 and q_2 of the two links.

Eqns. (A.1) and (A 2) for link 1 and link 2 respectively are,

$$f_{0,1} - f_{1,2} + m_1 g - m_1 \dot{v}_{c1} = 0 \qquad (A.3)$$

$$N_{0,1} - n_{1,2} = r_1\, C_1 \times f_{1,2} - I_i \dot{w}_i = 0 \qquad (A.4)$$

$$f_{1,2} + m_2\, g - m_2\, \dot{v}_{c2} = 0 \qquad (A.5)$$

$$N_{1,2} - r_1\, C_2 \times f_{1,2} - I_2\, \dot{w}_2 = 0 \qquad (A.6)$$

For the planar maipulator we have $N_{0,1} = u_1$ and $N_{0,2} = u_2$. Also, the angular velocities w_1 and w_2 can be expressed as $w_1 = \dot{q}_1$ and $w_2 = \dot{q}_2 + \dot{q}_1$ and the angular velocities v_{c1} and v_{c2} can be written as

$$v_{c1} = \begin{bmatrix} lc_1\, \dot{q}_1 \sin q_1 \\ lc_1 \dot{q}_1 \cos q_1 \end{bmatrix} \qquad (A.7)$$

$$v_{c2} = \begin{bmatrix} \{\, l_1 \sin q_1 + lc_2 \sin(q_1+q_2)\} \dot{q}_1 - lc_2 \sin(q_1+q_2)\, \dot{q}_2 \\ \{\, l_1 \cos q_1 + lc_2 \cos(q_1+q_2)\,\} \dot{q}_1 - lc_2 \cos(q_1+q_2) \dot{q}_2 \end{bmatrix}$$

$$(A.8)$$

Substituting eqns.(A.7) and (A.8) along with their time derivatives into eqns.(A.5)

and (A.6), we obtain the closed form dynamic equations in terms of q_1 and q_2:

$$\mathbf{D(q)}\,\ddot{\underline{q}} + H(\underline{q},\dot{\underline{q}}) + g(\underline{q}) = \underline{u}$$

Where $\underline{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$, $\underline{q} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$

$$D(\underline{q}) = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}, \quad H(\underline{q},\dot{\underline{q}}) = \begin{bmatrix} H_{11} \\ H_{21} \end{bmatrix} \text{ and } \quad g(\underline{q}) = \begin{bmatrix} g_{11}(q) \\ g_{21}(q) \end{bmatrix}$$

$$D_{11} = m_1 lc_1^2 + m_2(l_1^2 + lc_2^2 + 2\, l_1 lc_2 \cos(q_2) + I_2 + I_2$$

$$D_{12} = m_2 l_1 lc_2 \cos(q_2) + m_2 lc_2^2 + I_2$$

$$D_{21} = D_{12}$$

$$D_{22} = m_2 lc_2^2$$

$$H_{11} = -\, m_2 l_1 lc_2 \sin(q_2)\, \dot{q}_2^2 - 2\, m_2 lc_2 l_1 \sin(q_2) \dot{q}_1 \dot{q}_2 \qquad (A.9)$$

$$H_{21} = m_2 l_1 lc_2 \sin(q_2) \dot{q}_2^2$$

$$g_{11} = m_1 lc_1 g \cos(q_1) + m_2 g\,(lc_2 \cos(q_1 + q_2) + l_1 \cos(q_1))$$

$$g_{21} = m_2 lc_2 g \cos(q_1 + q_2)$$

APPENDIX B

B.1 Derivation of the f_1 and f_2 terms :

Consider the manipulator dynamic equation (2.2). The determinant of the $D(\underline{q})$ matrix representing the manipulator inertia is given by

$$|D| = m_1 m_2 lc_1^2 lc_2^2 + (m_1 m_2 lc_1 lc_2)^2 + m_2 lc_2^2 I_1 + m_1 lc_1^2 I_2 + I_1 I_2 - (m_2 l_1 lc_2)^2 \cos^2(q_2) \qquad (B.1)$$

$$\text{Now } \ddot{\underline{q}} = D^{-1}(\underline{q})[\underline{u} - H(\underline{q}, \dot{\underline{q}}) - g(q)] \qquad (B.2)$$

$$= \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

From eq. (B.2) it is clear that the f_1 and f_2 are given by the first and second rows of the matrix $D^{-1}(\underline{q})[\underline{u} - H(\underline{q}, \dot{\underline{q}}) - g(q)]$.

Using eqns (B.1), (B. 2) and (2.) we get

$$f_1 = \frac{1}{|D|}[c_1 u_1 - c_1 u_2 + c_2 \cos(q_1) + c_3 \cos(q_2) \cos(q_1 + q_2) + c_4 u_2 \cos(q_2) + c_5 \sin(q_2)(\dot{q}_2^2 + 2\dot{q}_1\dot{q}_2) + c_5 \sin(q_2)\dot{q}_1^2 + c_6/2 \sin(2q_2)]$$

$$= \frac{f_{1u}}{|D|}$$

where

$$c_1 = m_2 lc_2^2 + I_2$$

$$c_2 = -c_1 g (m_1 lc_1 + m_2 l_1)$$

$$c_3 = m_2^2 l_1 lc_2^2 g$$

$$c_4 = -m_2 l_1 lc_2$$

$$c_5 = -c_1 c_4$$

Similarly the expression for f_2 is given by

$$f_2 = \frac{1}{|D|}[-c_1 u_1 + c_4 u_1 \cos(q_2) + c_7 u_2 - c_2 \cos(q_1) + c_6/2 \sin(2 q_2) (\dot{q}_2^2 + 2\dot{q}_1\dot{q}_2 + \dot{q}_1^2) + c_5 \sin(\dot{q}_2^2 + 2\dot{q}_1\dot{q}_2) + c_8 \cos(q_1) \cos(q_2) - c_9 \cos(q_1 + q_2)\cos(q_2) + c_9 \cos(q_1 + q_2) + c_{10} \sin(q_2) \dot{q}_1^2$$

$$= \frac{f_{2u}}{|D|}$$

where

$$c_7 = m_1 lc_1^2 + m_2 l_1^2 + m_2 lc_2^2 + I_1 + I_2$$

$$c_8 = m_2 l_1 lc_2 g (m_1 lc_1 + m_2 l_1)$$

$$c_9 = m_2 lc_2 g (c_1 - c_7)$$

$$c_{10} = -c_7 m_2 l_1 lc_2$$

B.2 Expressions for the Jacobian Matrix Elements :

$$\frac{\partial f_1}{\partial q_1} = \frac{1}{|D|} [-c_2 \sin (q_1) - c_3 \cos (q_2) \sin (q_1 + q_2)]$$

$$\frac{\partial f_1}{\partial q_2} = \frac{1}{|D|^2} [f_{1u}' |D| - f_{1u} |D|']$$

where f_{1u}' and $|D|'$ are given by

$$f_{1u}' = \frac{\partial f_{1u}}{\partial q_2} = c_3 [-\sin (q_2) \cos (q_1 + q_2) - \sin (q_1 + q_2) \cos (q_2) + c_4 u_2 \sin (q_2) + c_5 \cos (q_2) (\dot{q}_2^2 + 2 \dot{q}_1\dot{q}_2 + \dot{q}_1^2) + c_6 \cos (2q_2) \dot{q}_1^2]$$

$$|D|' = \frac{\partial |D|}{\partial q_2} = (m_2 lc_2 l_1)^2 \sin (2q_2)$$

$$\frac{\partial f_1}{\partial \dot{q}_1} = \frac{1}{|D|} [2 c_5 \sin (q_2) (\dot{q}_1 + \dot{q}_2) + c_6 \sin (2q_2) \dot{q}_1]$$

$$\frac{\partial f_1}{\partial \dot{q}_2} = \frac{1}{|D|} [2 c_5 \sin (q_2) (\dot{q}_1 + \dot{q}_2)]$$

$$\frac{\partial f_2}{\partial q_1} = \frac{1}{|D|} [c_2 \sin (q_1) - c_8 \sin (q_1) \cos (q_2) - c_9 \sin (q_1 + q_2) + c_3 \cos (q_2) \sin (q_1 + q_2)]$$

$$\frac{\partial f_2}{\partial q_2} = \frac{1}{|D|^2} [f_{2u}' |D| - f_{2u} |D|']$$

where f_{2u}' is given by

$$f_{2u}' = \frac{\partial f_{2u}}{\partial q_2} = [(c_4 u_1 + 2 c_4 u_2 - c_8 \cos (q_1)) \sin (q_2) - c_6 \cos (2q_2) (\dot{q}_2^2 + 2 \dot{q}_1\dot{q}_2 + \dot{q}_1^2) - c_9 \sin (q_1 + q_2) + c_{10} \cos (q_2) \dot{q}_1^2 + c_3 \sin (2q_2 + q_1) - c_5 \cos (q_2)(\dot{q}_2^2 + 2 \dot{q}_1\dot{q}_2)]$$

$$\frac{\partial f_2}{\partial \dot{q}_1} = \frac{1}{|D|} [- c_6 \sin (2q_2) (2 \dot{q}_1 + \dot{q}_2) + 2 c_5 \sin (q_2) \dot{q}_2) + 2 c_{10} \sin (q_2) \dot{q}_1$$

$$\frac{\partial f_2}{\partial \dot{q}_2} = \frac{1}{|D|} [(-c_6 \sin (2q_2) + 2 c_5 \sin (q_2) (\dot{q}_1 + \dot{q}_2)$$

$$\frac{\partial f_1}{\partial u_1} = \frac{1}{|D|} c_1$$

REFERENCES

[1] R Horowitz and M Tomizuka, "*An Adaptive Control Scheme for Mechanical Manipulators - Compensation of Nonlinearity and Decoupling Control* ", ASME. Journal of DSMC, Jan 1986, Vol 108, pp 127-135 .

[2] S Arimoto and Takegaki, " *An Adaptive Method for Trajectory Control of Manipulators* ", 1981, Proc. of 8 th IFAC World Congress, Kyoto pp 1921-1926 .

[3] C.S.G.Lee and M.J.Chung, " *Adaptive Perturbation Control with Feedforward Compensation of Robot Manipulations* ", Simulation, 1985, Vol.44,No. 3 pp 127-136.

[4] M.H.Liu, " *An Adaptive Control Scheme for Robotic Manipulators* ", Proc. of IEEE Int.Conf.on Robotics and Automation, 1987,Vol3, pp 1386-1391.

[5] P.V.Waknis, " *On Adaptive Control of Robots*", M.Tech. Thesis, Dept. of Elec. Engineering, I.I T Kanpur, 1988.

[6] A.Ghosh, B.R Patnik and P V.Waknis, " *On Adaptive Control of Robot Manipulators Based on Dynamic Parameter Estimation* " , Proc. of 4th Int. Conf on CAD, CAM, Robitics & Factories of Future ; Vol 1, pp 591-607

[7] A.Balestrino, G.De.Mano and L.Sciaricco, " *Robust Control of Robotic Manipulators* ", Proc. of IFAC, Vol. 5, 1985, pp 2435-2439.

[8] Y.Oshima and M.Miyasato, " *Non Linear adaptive Control of Robotic Manipulators With Continuous Control Inputs* ", Int.J.of Control, 1989, Vol 49 pp 545-549.

[9] P.Hsu, S.Sastry and M.Bodson and B.Paden, " *Adaptive Identification and Control of Manipulators With Joint Acceleration Measurements*", Proc. of IEEE Int. Conf. on Robotics and Automation, 1987, Vol 2 .

[10] R.H Middleton and G C Goodwin, " *Adaptive Computed Torque Control for Rigid Link Manipulators*", Systems & Control Letters, 1988, Vol10, pp 9-16.

[11] J.E.Slotine and W.Li, " *Adaptive Manipulator Control - A Case Study* ", Proc. of IEEE Int. Conf. on Robotics and Automation, 1987, Vol3, pp 1394-1400.

[12] W.Li and J J.Slotine, " *An Indirect Adaptive Robot Controller* ", Systems&Control Letters, 1989, pp 9-16.

[13] A.J.Goldenberg, J.A.Apkarian and H.W.Smith, " *An Approach to Adaptive Control of Robot Manipulators Using Computed Torque Technique*", ASME J.of DSMC, March 1989, Vol 111, pp 1-8.

[14] M.Blauer and P.R. Belanger, " State and Parameter Estimation for Robotic Manipulators Using Force Measurements" , IEEE Trans. on Automatic Control, Vol. AC-23, No 12, December 1987.

[15] S.Majee, " *Control of Robot Manipulators under Uncertain Dynamics and Measurement Errors* ", M Tech. Thesis, Dept of Elec. Engineering, I.I.T Kanpur, 1989.

[16] H.Asada and J.J.E Slotine, " *Robot Analysis and Control* ", John Wiley Intl Newyork, 1986 .

[17] M.Trakoh, " *Simulation Analysis of Dynamics and Decentralized Control of Robot manipulators* ", Simulation, Vol. 6, October 1989, pp 169-176.

[18] E.A. Misawa and J.K. Hedrick, " *Nonlinear Observers -- A State-of-the-Art Survey* ", ASME J of DSMC, Vol. 111, September 1989, pp 344-352.

[19] F.L.Lewis, " *Optimal Estimation : with an Introduction to Stochastic Control Theory* ", John Wiley & Sons, 1986.

[20] M.G.Safonov and M.Athans, " Robustness and Computational Aspects of Nonlinear Stochastic Estimators and Regulators " , IEEE Trans. on AC, Vol

28, 1978, August 1978, pp 771-725

[21] A.Ghosh and S. Majee, " *Filtering of Noisy Measurements Under Uncertain Robot Dynamics* " , INDO-US Workshop on Spectral Analysis in One or Two Dimensions

[22] C.G.Atkeson, C.H.An and John. M. Hollerbach, " *Estimation of Inertial Paraeters of Manipulator Loads and Links* ", Proc of The 3 rd Int. Symp. on Robotics Research, 1986, pp 221-227.

$$\frac{\partial f_1}{\partial u_2} = \frac{1}{|D|}\left[-c_1 + c_4 \cos (q_2) \right]$$

$$\frac{\partial f_2}{\partial u_1} = \frac{1}{|D|}\left[-c_1 + c_4 \cos (q_2) \right]$$

$$\frac{\partial f_1}{\partial u_2} = \frac{1}{|D|}\left[c_7 - 2\, c_4 \cos (q_2) \right]$$

EE-1990-M-KOM-ROB.